AN INVESTIGATION OF AN AUTOMATIC MACHINE GRADING SYSTEM FOR ASSEMBLY LANGUAGE INSTRUCTION

by

Roger Francis Bacon

United States
Naval Postgraduate School

THESIS

AN INVESTIGATION OF

AN AUTOMATIC MACHINE GRADING SYSTEM

FOR ASSEMBLY LANGUAGE INSTRUCTION

by

Roger Francis Bacon

June 1969

An Investigation

of

An Automatic Machine Grading System

for

Assembly Language Instruction

by

Roger Francis Bacon
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1959

Submitted in Partial Fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1969

ABSTRACT

Machine grading techniques are becoming of greater importance because of the increased number of students in programming classes. Characteristics and limitations of automatic machine grading systems are proposed. A grader for introductory assembly language programming courses was developed. The properties of this grader are discussed and an example of a graded program is given.

TABLE OF CONTENTS

LIST OF TABLES

ACKNOWLEDGEMENTS

The author wishes to acknowledge the enthusiastic help given by G. L. Barksdale, Jr., of the Naval Postgraduate School. The author is also indebted to Professor Andries van Dam and Richard M. Kogut of Brown University for the opportunity to use the Brown University Student Operating System.

I. INTRODUCTION

A. BACKGROUND

The capability of an instructor to teach programming courses and to grade student programs in classes with large numbers of students has been significantly enhanced by automatic machine grading systems. The concept of computer-controlled grading of student programs was first documented by Hollingsworth [1] in 1960. Machine grading systems are known to exist at Rensselaer Polytechnic Institute [1], Stanford University [2,3], and at universities in England and Australia. These grading systems provide a method of objectively grading student programs automatically for the instructor. By utilizing the computer as an extension of his teaching methods, an equal amount of attention in evaluating class programming work is possible.

B. OBJECTIVE

The objective of this thesis is to investigate the concept of machine grading systems and to provide a practical implementation of a grader for beginning assembly language students. The fulfillment of this objective is accomplished by an analysis of grading techniques and parameters available for evaluating a program. The second element of this objective is accomplished by establishing an operating system specifically designed for assembly language instruction at the Naval Postgraduate School and constructing a machine grading program which will operate within this system and produce graded student programs. Through the courtesy of Professor Andries van Dam of Brown University, the Brown University Student Operating System was made available to the Naval Postgraduate School for use in the instruction of assembly language.

C. METHODOLOGY AND ORGANIZATION OF THESIS

The methodology of this investigation was to first review previous work in the field of machine grading. The metrics of a program which can be utilized in its evaluation are discussed in Chapter II. A summary of the essential properties of an automatic grading system is presented at the end of Chapter II. The description of a grading system and its implementation is given in Chapter III. In addition, grader use with student programs is discussed. The conclusions reached after the investigation, and the recommendations for further study are contained in Chapter IV.

The first step in constructing a grader system was to revise the structure of the Brown University Student Operating System and implement it at the Naval Postgraduate School. The Student Operating System was then examined for inherent characteristics which provide the instructor with teaching and grading capability. An appraisal of this system with respect to grading capabilities is presented in Appendix A. The other appendices contain various computer programs, flowcharts, and outputs relating to this study.

II. CHARACTERISTICS OF A GRADER

A. REVIEW OF AUTOMATIC GRADING SYSTEMS

A search of the literature for reports of automatic grading techniques produced several documents specifically addressing the subject. Perhaps the first published report of grader programs is that of Hollingsworth [1]. This grader was used at Rensselaer Polytechnic Institute in 1959 and was implemented on an IBM 650 computer. The author states that the university could not have accommodated programming classes of 80 to 120 students without the use of the grader. This early version of an automatic grader provided batch processing and grading of student jobs for a variety of exercises.

Students submitted new programs and corrections to old programs for keypunching at the end of a class period. After keypunching, the corrections were filed with the old programs and all the programs, old and new, were arranged by exercises. The grader program deck was inserted with the student jobs. Considerable effort in hand filing punched cards was necessary in this system.

This grader punched identification cards for student programs, checked results and punched either WRONG ANSWER for an incorrect solution or PROBLEM COMPLETE for all correct answers.

A great deal of operator control was required to process student programs. Manual re-entry for overflow, invalid addresses, and other problems was required. A limitation of this grader was the fact that it processed only machine language programs. Student programs could modify the grader itself, causing additional problems.

The grading system required the specification of four addresses associated with the exercise, viz., the location of the entrance to the student program, the location to which the student program must exit, the address of variables and the address of results.

An advantage of this system was that programming classes were not required to work on a single project at the same time. Since various exercises could be graded during one batch run, students could work on several projects simultaneously if they so desired. Despite some difficulties with this system, Hollingsworth [1] indicated that the grader had more than justified the effort in its implementation.

Automatic machine grading programs in use at Stanford University are reported by Forsythe and Wirth [2,3]. Two ALGOL grading programs have been used in connection with introductory ALGOL programming and numerical analysis courses at Stanford University since 1961. One grader simply furnishes data and checks answers. The other program provides a searching test of the reliability and efficiency of an integration procedure. The major difficulties and limitations of the Rensselaer grader were overcome in the grading systems in use at Stanford. The concept of basing a grade only on the binary answer "CORRECT" or "WRONG" was used for beginning students. More advanced students were graded by the second grader on their ability to solve the integration problem efficiently and with some degree of reliability. The authors claim that this grader marks a further step in the automation of grading because the quality of the program is evaluated.

R. E. Berry [4] examined the use of automatic grading programs for checking the practical work of students in a numerical analysis

course. This report provides a comparison of the Forsythe and Wirth program [2] and a grader program by Naur [5]. Both of these graders considered the same problem (finding the root of a given function to a given accuracy in a prescribed interval).

Naur's method required that the student program be included in the grader program as a labelled block of coding, accessed by means of a switch. The grader assigned values to variables which were global to the block. In this manner different problems could be presented to the student algorithm. When a solution was achieved, a branch to a given label within the grader program was made and an examination of the result was conducted. Further problems were then presented until the problem list was exhausted and the grader continued with the next student program. In contrast, the ALGOL grader implemented at Stanford [2] was called by the student program. The students were required to write a procedure with parameters in a specific order. Each procedure was tested by supplying it with different sets of parameters and checking the solution obtained.

One of the difficulties of grading was well illustrated in the work by Berry [4]. The problem of evaluating the quality of an algorithm for a problem which has various methods of solution is exceedingly difficult. The author states that in trying to present a problem to which different methods of solution are available, the possibility of successfully grading the work presented is remote.

Berry points out that the demand for automatic grading programs in Great Britain is not expected to be substantial, since computer classes are smaller. One commendable aspect of graders was verified by both Naur [5] and Berry [4]. They arouse interest and provide

an incentive to work. Students submitting programs to this type of grader found the challenge of writing the best procedure an incentive in their work.

The first known grading procedure for PL/1 exercises was implemented on an IBM System/360 model 50 computer at the Australian National University in 1966 [6]. The decision to use automatic grading by computer of all student attempts at class problems was based on the following:

1. The large volume of work required for teachers to carry out grading and associated record keeping.

2. The apparent difficulty of teaching input/output sufficiently early in the course for students to begin writing programs.

3. The ease with which input test values could be supplied to the students and results printed under control of a grading procedure.

The idea of using a grading system to bypass difficult programming aspects during beginning phases of instruction was not found in earlier documentation of graders. Temperley and Smith [6] also point out that the use of PL/1 made possible the evaluation of results in a variety of formats. Test data and results could be in the form of arrays, structures, bit-strings and character-strings as well as scalar numeric values.

The timeliness of this investigation is borne out by a recent paper on an automatic grading scheme for simple programming exercises by Hext and Winings [7]. This report documents the development of the Basser Automatic Grading Scheme (BAGS) at the University of Sydney, Sydney, Australia. A major difference in the BAGS system is that it can handle exercises in several different languages. No special

action by computer operators is required with this scheme. Another significant departure from previously documented graders is the structure of the system itself. Whereas most other schemes embed their exercises in some larger program, BAGS is part of the standard operating system and its exercises are run under batch processing methods.

The basic requirements of the program were as follows:

" (1) It should handle exercises in ALGOL, in MINIGOL (a subset of ALGOL) and in KDF 9 Assembly Code.
(2) It should not place any additional burden on the operators.
(3) It should record every attempt at an exercise, with sufficient data for calculating a mark.
(4) It should provide summaries on request for specified classes and exercises over a given period." [7]

The method of grading in the BAGS system is unique. The assignment of credit for an exercise not only takes into account the mark for each program attempted but also the number of attempts to solve the problem. The maximum mark for a single exercise was set at five. One point was credited for each of the following:

a. The program compiled successfully.

b. The program ran to completion.

c. The first answer was correct.

d. The second answer was correct.

e. The program ran successfully within a prescribed central processor time.

The graded results were printed for each student in the form of i/j. This grade states that the student made i attempts at the exercise and that his best mark was j. If the student's program satisfied all five criteria the first time, a mark of 1/5 was assigned. A grade was assigned by the formula:

$$\text{GRADE} = \text{average of } \frac{100 \times j}{4 + i}$$

The average was taken over all exercises being marked.

The original version of the Brown University Student Operating System included a grading system. The SOS grader has subsequently been eliminated from the SOS system at Brown University. In a search for properties of a grader it is revealing to analyze the characteristics of the original SOS grader to determine the cause of its demise. The operation of the original SOS grading system is described as follows:

> "Using appropriate supervisor calls, the student communicates answers and messages to the grader to indicate the problem to be graded (initialization), the answers and error exits to be generated, and to obtain special information dependent on the specific program being graded. In addition to communicating with the grader while his program is being executed, the student must write his program following certain conventions: 1) various sections of his program must bear standard labels as indicated in each problem assignment; 2) the data is read in the usual manner, but must be placed in an assigned location; and 3) the expected answer must be in the location and form as specified in each problem assignment." [10]

As can be seen by this description a great deal is required of the student to ensure that his program is processed properly by the grader. After attempting two different versions of the grader at Brown University, it was found that the student programs were required to initiate almost all communication with the grading program and doing this made the student programs both more difficult to write and more difficult to make efficient, due to the rigidity of the communication conventions. A second, and fatal, limitation of this system was the inability to specify a programming problem explictly enough to allow the student to follow grader conventions without actually specifying the method of solution. A review of two student programs which had been graded by this system and

discussions with the authors of the system verified these problem areas as the reasons for elimination of the grading feature from the Student Operating System at Brown University.

It is evident that significant improvements in automatic grading techniques have evolved since the development of the Rensselaer grader in 1959. In the previous discussion of work in this field an attempt has been made to chronologically illustrate this evolution. Reference 7 states that 1500 first-year students will be enrolled in programming classes at the University of Sydney in the next two terms. Under these circumstances, the value of grading systems seems well established. However, much of the documentation on this subject has concerned implementation of a particular grading program. The next section proposes certain metrics of an algorithm which might be used in a grading system. The effectuation of some of these ideas may not be practical at present. Nevertheless, with the increased interest in automatic grading an exploration of conceivable characteristics of a grader is deemed necessary.

B. THE METRICS OF AN ALGORITHM

The term Grader Program is perhaps a misnomer; not all such programs assign a mark or grade to the work submitted. A grading system is sufficient if the program can provide enough information for a mark to be quickly assigned by the student's supervisor. The basis for the output of this information is discussed in the subsequent paragraphs.

It is advisable to discuss the upper and lower bounds of a grading system's performance. For instance, it is a well known fact that there is no algorithm to determine whether or not a given procedure

is total. That is, a grader program cannot determine whether another program contains no loops. An upper limit in the performance of a grader program might be the class of functions which fall in this category. A second class of characteristics of a grader which may limit achievable performance, is the class of functions which are computable but are impractical to implement. Berry [4] has shown that considerable complication results in grading a problem in which any one of several methods of solution might be acceptable. A lower bound in the performance of a grader is simply that it must produce some information about the algorithm being tested. A program which recorded the trivial fact that a student program had been processed provides information which might be used to evaluate a student's progress even though the program itself is not evaluated. Therefore, it can be concluded that there exists a range in which a grader must operate to produce the basis for the evaluation of an algorithm.

The most common measure used in the evaluation of a student exercise is the determination of whether or not the solution is correct. This measure may provide a sufficient basis for a grade for introductory programming exercises [1]. However, it is suggested that this is dependent on the characteristics of the problem assigned. A grader should be capable of providing information pertaining to the accuracy of a program result. In addition, a grading program should be capable of measuring this value in a variety of formats such as in the BAGS grading system [7].

Criteria for evaluating a program which have been used are successful assembly, successful execution, and the number of

attempts at an exercise [7]. Such criteria form an absolute judgement of a program in that the evaluating process does not consider the quality of the program.

A grader which only evaluates the results of another program does not provide specific information about the program itself. Qualities of a program which have significance with respect to grading are execution time, storage utilization, programming techniques, program logic and annotation. These characteristics form a basis for the measure of the quality of a program.

The completion of a programming exercise within a sufficiently short central processor time was one criteria in the BAGS method of grading [7]. This measure of a program can provide the instructor with a means of evaluating program efficiency. Application of this measurement is more appropriate for advanced programming classes.

The amount of storage utilized by a program is a characteristic which can be used to evaluate its quality. This measure is all too often neglected in programming classes but has direct application in the field. Incorporation of this measurement in a grading system can be an effective means of stressing the importance of the efficient use of storage.

If an instructor compared a program which contained one hundred ADD instructions to sum one hundred numbers with a program which used a single indexed ADD instruction in a program loop, the latter would undoubtedly be considered superior. Although programming techniques vary considerably, the quality measure of a program may provide the best criteria for evaluation of a programmer's efforts. Due to the fact that good programming techniques decrease execution time and

provide efficient use of storage, this measure is more effective than analysis of time or storage parameters alone. The multiplicity of thought forms a substantial barrier to the implementation of this characteristic in a grading system. A specific problem could possibly be designed which then would be inspected for key variables in a prescribed optimum order by a grading system. It appears that the implementation of such a system would require the development of an extremely capable heuristic pattern matching algorithm. It is doubtful whether the effort would be justified in obtaining this measurement.

A definition of program logic is given by Montalbano [8]. The subject matter of program logic is the matching of appropriate actions to given conditions. For example, program logic is concerned with such statements as "if a given record is an inventory adjustment, process the record before all other records affecting this stock item; if the record is a receipt, process it before any sales order affecting this stock item." In essence, program logic is the effective description of computer outputs as functions of inputs. Techniques such as narrative description, program comments, flow charts, decision tables and system specification languages of the Iverson type are tools for expressing program logic [8].

The logical structure of an algorithm is another measure which forms a basis for the evaluation of the program. The evaluation of the logic of the program is extremely difficult because of the variety of paths which a programmer may take to solve the problem. Chapter IV describes the implementation of a grader which can evaluate subsections of a program independently. The logical

order in which a solution was attempted might be measured in the broadest sense by verifying the result of each sequential section of a program. A method to compare program logic descriptions such as those described in [8] would be an invaluable grading technique. At the present time an inspection by the instructor of program logic descriptions such as flow charts is required.

Higher level languages have an advantage over assembler languages in that they possess a "self-documenting" feature [9]. The documentation of assembly language programs is an important area which requires evaluation. The annotation of a program varies from individual to individual. The basic rules of documentation for assembly language programs are described by Opler [9]. It is doubtful if a machine process could automatically produce an evaluation of a student program with respect to these rules. This author believes that graphic methods may prove to be an effective way of quickly evaluating the logic and the documentation of a program. A correct logic flow chart might be superimposed on a student flow chart on a display device. Key decision blocks could then be checked by the instructor. Spot checking of annotation can be accomplished on a graphic display device if the student programs are disk resident such as in the SOS system (Appendix A).

The concept of providing automatic methods which will evaluate the quality of a program is in its infancy. It is expected that further work in this field will incorporate some of the previously listed metrics of a program.

C. GRADER DESIGN CONSIDERATIONS

The first consideration in designing a grading system should be the specification of objectives. What type of information is necessary to evaluate the programs in a particular course? The designer must concern himself with the previously mentioned parameters of a program which are required to be measured.

The second step in the design of a grader is to carefully investigate the system available for grader use. Forsythe [3] reports that the Stanford grader was possible only because of the well-designed BALGOL compiler with its own compiler-with-library generator, and because a procedure called BUTTERFLY could generate relocatable machine-language programs. It was found that neither the IBM 7090 BALGOL system nor the Burroughs B5000 ALGOL 60 system were well adapted to the grading problem at Stanford University.

One of the most critical characteristics in the design of a grading system is the amount of involvement of the student with the grader itself. The student should not be required to alter his program significantly to conform to grader conventions. A minimum of communications between the grader and the student, with maximum communication capability between the instructor and student should be the rule in a grading operation.

Another feature of a grading system which must be considered is the work required by the instructor to program the grader to process the specific problem. The grading system should include macro-instructions for instructor messages and general communication sections for the passing of answers and data. In general, instructor involvement with the grader should be held to a minimum.

The student should be able to work out solutions using his own data, but data should be provided by the instructor in some manner at grading time. Grading of the program with data from the instructor could present situations which had not occurred to the student.

A grading system should be able to handle a variety of problems. Programming a complete grading system to evaluate a specific problem would be inefficient. Manual intervention by computer operators to process grading jobs is impractical on present large scale computing systems such as the IBM/360. Execution time of the student program should not be significantly increased when executed for grading. Additional characteristics such as the ability to evaluate programs written in different languages, and the capability to change problem parameters with minimum effort should be considered in the design of a grading system.

A summary of proposed grader design characteristics is presented in Table I. This list is a summary of features which might be considered in the design of a grading system.

These characteristics of a grading system have been proposed as the result of study of documented grading systems and also from limited experience gained in implementing a practical grader. Such features, if only achieved to some minimum degree, will provide a grading system which can assist the instructor greatly in programming instruction classes. The next chapter discussed the mechanization of a grading system which possesses most of these characteristics and is primarily designed for use in beginning assembly language programming courses.

TABLE I

SUMMARY OF GRADER DESIGN CHARACTERISTICS

A. EVALUATION CAPABILITY

CHARACTERISTIC EVALUATED	TYPE OF MEASURE
Result of Algorithm	Absolute
Tolerance of Solution	Absolute
Successful Assembly	Absolute
Successful Execution	Absolute
Number of Attempts at Exercise	Absolute
Assembly Time	Quality
Execution Time	Quality
Number of Instructions Executed	Quality
Storage Utilization	Quality
Program Logic	Quality
Method of Solution	Quality
Documentation	Quality

B. MINIMUM STUDENT COMMUNICATIONS WITH GRADER

C. MINIMUM INSTRUCTOR EFFORT IN USING GRADER

D. CAPABILITY TO GRADE DIFFERENT LANGUAGES

E. CAPABILITY TO GRADE A VARIETY OF PROBLEMS

F. AMOUNT OF OPERATOR ATTENTION REQUIRED

G. EFFORT REQUIRED TO ALTER OR ADD PROBLEMS TO GRADING SYSTEM

H. DATA MANIPULATION FOR PROBLEM DURING GRADING RUN

III. IMPLEMENTATION OF A GRADER

A. PURPOSE OF GRADER

Since the Student Operating System (Appendix A) is well-suited for beginning assembly language programmers, a grading program for use in introductory programming classes was written. The primary purpose of this program is to check student answers and assign a grade. It is designed to be used for grading small projects assigned to students who are just learning the language. With a grading system to handle numerous small projects, concentration on one particular area such as indexing, looping or arithmetic operations is possible and a project can be assigned to cover that subject. This type of grader allows the instructor to assign more projects and gives the student an opportunity to obtain machine experience on a wider variety of assembly language programming aspects. This would not be permissable with a system which was designed for a specific problem of a more complex nature.

The grader is an absolute type of system in that it operates on answers from the student. In addition to checking answers, communications from the instructor to the student have been included as an essential part in the design of this grader. In summary, the purpose of the grader is to allow the instructor to assign numerous small projects in an introductory assembly language course, to grade the projects and provide messages from the instructor to the student. By using the computer as a tool for evaluation of class work, the instructor can provide an equal amount of attention to appraising each student's progress.

B. MECHANIZATION OF THE GRADER

Although the Brown University Student Operating System Grader program has been eliminated from the system, the structure for including a grader was intact in the system received. By choosing the GRADER=YES option at system generation time, communication links, control blocks, and initialization of parameters for a grader were generated within the SOS system. Through the use of parts of this available structure and modifications to o. elimination of some of its features, it was possible to concentrate efforts on the grading program itself.

The grader operates under control of the SOS control program, and is called by the student when he feels that his program is correct. The basic structure of the grading program and its interrelation with the rest of the SOS system is shown in Chart Number 1, Appendix C. During initialization of a student's program the SOS control program calls the bookkeeper. The bookkeeper sets system options specified by the user. If the student has specified GRADER=YES on his job card, grader parameters are initialized by the bookkeeper and the grader program is called.

A heading indicating that this is a grader run is printed and the SOS interpreter is then called from the grader. The SOS interpreter executes SOS machine code interpretively. The first two instructions of the student's program are executed setting up initialization of the grader for the particular problem specified by the student.

After grader initialization, the interpreter continues execution of the student's program until it is desired to have an answer checked. The student issues a supervisor call and indicates

the location of his answer. The main section of the grader checks the answer and assigns a grade, substitutes the correct answer if the instructor so desires, prints the results and any messages from the instructor if the answer was not correct. Control is passed to the interpreter again for further execution of the student's program. This technique of grading the student's program during execution continues until the final answer is graded. Return to the control program is then made and the next student job is processed.

The grading program has an initialization section to initialize the grader for a specific problem. The main section of the grader handles all aspects of grading the problem. A communications section for passing answers and messages from the instructor is included. The grader is designed to grade up to five intermediate answers for a problem and a final answer. The capability of grading twelve different problems is included in the grading program. Six different answers should suffice as a maximum for beginning programming projects. The capability to grade twelve different problems allows the instructor to assign one problem per week during the class quarter.

The first part of the grading program establishes addressability with the other major sections of the SOS system and saves the return address to allow return to the control program at the end of the grading run. A grader heading is printed and the interpreter is given control. Exeucution of the first instruction in the student's program starts the initialization of the grader. The return address to the interpreter is first saved and the program control counter is incremented to point to the next sequential instruction in the student program. A check for student core wrap around is made and a

check for a proper initialization call is accomplished. A halfword grader control block is used throughout the program to set various conditions. The first byte of this control block is used by the system. The bookkeeper sets a bit on if the student has asked for a grading run. The grader initialization section turns a second switch on if grading is in process. The second byte of the grader control block is used by the instructor to control various functions of the main grading section and its use will be described in the section on use of the grader.

If a grading run has been requested and initialization has not yet taken place, the data for the problem specified is placed in a general communication section. This is accomplished by the use of a dummy section which describes the layout of storage for the data that the instructor has defined for a specific problem. Symbolic names defined in the dummy section are then used as operands in the main grading section during the grading process. This method allows the instructor to separately assemble the appropriate data in a control section and use the link editor to link his answer CSECT in object form to the system.

Various error routines and messages are included in the grader initialization section. Upon completion of initialization, a message to that effect is printed and execution of the student's program is continued. The grading program grades the result of up to five intermediate sections which the student has used in his program to achieve a final answer to a specific problem. When the student has reached a particular result, the main grader section of the grading program is called. This section first determines which

answer has been specified, and then branches to the appropriate routine. The grading routines for intermediate answers are similar except for correct answer usage data and the messages sent to the student.

The intermediate answer grading routines first obtain the student's answer. This is accomplished by the subroutine called GETANS. The address of the answer in student core, which is word-oriented, is obtained and converted to a System/360 address. The correct answer specified by the instructor is then compared with the student's answer. If the answer is correct, a message indicating this fact is printed and credit for that answer is accumulated. The accumulated credit for each intermediate answer is also printed. If the student answer is not correct, a test is made to see if the instructor has a message for the student which might assist him in obtaining the correct answer. If the instructor has included a message, it is printed on the student's output. The instructor has the option of substituting correct data for a wrong answer if he so desires. Thus the subsections of a student program can be checked even if the program as a whole computes incorrectly. If the instructor has chosen this option, the correct data is moved into the student core address of his incorrect answer. No credit is accumulated for a wrong answer. A message indicating that the answer was not correct is printed.

If there is just one answer for a particular problem or if a final solution has been reached, the final answer routine of the main grading section is entered. The final answer routine checks the solution and prints a message indicating whether it is correct or not. A total grade is then determined. Fifty points are given

for a correct final answer. Each of the five intermediate answers are worth ten points. A grade for a problem with five intermediate answers and a final answer is totaled by summing the accumulated intermediate answer credit and the credit for the final answer. If the problem required less than five intermediate answers, the accumulated credit for the intermediate answers is first supplemented to fifty and then added to the credit for the final answer. The assignment of these credit values for intermediate and final answers is arbitrary and may be changed by an instructor with only minor modifications to the grading program. The total machine grade and any final message to the student is then printed on the student's output.

The next part of the grading program evaluates the efficiency of the student's program to some extent. The number of instructions executed in the student program is printed. The mean number of instructions executed in previous programs for this project is supplied by the instructor in his answer control section. These two values are compared and additional credit is given to the student whose program required the execution of fewer instructions. A message indicating whether the student required more or less than the median is printed. The total machine grade with bonus credit is then printed. This capability is optional and is set by the instructor in the answer control section.

A return to the SOS control program is then made for further processing of student jobs. The grader program is included in Appendix B. Flowchart documentation of the program is included in Appendix C.

C. USE OF THE GRADER

This section describes the conventions which must be followed by the student and the instructor to ensure proper operation of the grading system. Examples of graded student programs are shown in Appendix D. The control section established by the instructor to communicate answers and messages to the grader is also included as Appendix E.

1. Student Grading Conventions

The student is required to indicate four items to the grader. He must first specify that he desires his program to be graded. The student then indicates which problem is to be graded. When the student has reached a solution to the problem he must indicate to the grader which of the six possible answers he wants to have checked and the location of that answer.

a. Calling for a Grading Run

When the student feels that his program is ready for grading, he utilizes the JCL bookkeeping parameter option feature of the SOS system to call the grader. This is done by placing GRADER=YES on the job card starting in column 26.

EXAMPLE:

Column:	1	6	16	26
	/JOB	BACONRF	SOSJOB1	GRADER=YES

b. Specification of the Problem to be Graded

To indicate which problem is to be graded the student places as the first executable instruction of his program the supervisor call SVC 3,0 and a second instruction DC (decimal number 0 through 11) immediately following the supervisor call. The supervisor call initializes the grader for the problem specified in the

second instruction. For example, if the student desired problem number three to be graded, the first two instructions of his program would be as follows:

Column:	10	16
	SVC DC	3,0 2

c. Specification of the Answer to be Graded

The student must indicate to the grader which answer is to be graded. This is accomplished by issuing a supervisor call.

EXAMPLE:

SUPERVISOR CALL	SPECIFIES THIS ANSWER
SVC 3,1	First intermediate answer
SVC 3,2	Second intermediate answer
SVC 3,3	Third intermediate answer
SVC 3,4	Fourth intermediate answer
SVC 3,5	Fifth intermediate answer
SVC 3,6	Final Answer

d. Specifying the Location of the Answer

Immediately following the supervisor call specifying the answer, the student places a HALT instruction with a single operand which is the symbolic name of the location of the answer.

An example of a student requesting a check of his fourth intermediate answer which had been previously stored in a storage location called ANSWER would be as follows:

Column:	10	16
	SVC H	3,4 ANSWER

2. Instructor Grading Conventions

The instructor must indicate four items to the grading system: the number of answers to be graded; the correct answers; messages to the student; and grader options.

a. Specification of Answers

To pass correct answers to the grader, the instructor defines fullword constants through the use of a macro-instruction called ANS. The answers will be defined in the instructor's control section.

EXAMPLE:

Column:	10	16
	ANS	8,13,-99,33397

The above example defines three intermediate answers and a final answer. Answers must be specified in order.

b. Specification of the Number of Answers

The number of answers to be graded is defined as a fullword constant in the control section.

EXAMPLE:

Column:	1	10	16
	NOINTANS	DC	F'decimal number 0-5'

c. Specification of the number of instructions executed.

The instructor specifies a value for the grader to use in the comparison of instructions executed by the student's program by defining a fullword constant in the control section.

EXAMPLE:

Column:	1	10	16
	NOX	DC	F'decimal number'

d. Specification of grader options

The grading system has three optional features. Messages to the student, the substitution of correct answers, and the assignment of additional credit are chosen by the instructor through the use of a macro-instruction called PROF. If it is desired to include all three features the default is chosen as follows:

Column:	10	16
	PROF	(blank operand field)

Operands to eliminate these features are:

NOMSG - eliminates message feature
NODATA - correct answers are not substituted
NOBONUS- feature to check the number of instructions executed and assign bonus credit is removed

EXAMPLE: To prevent the substitution of correct answers and eliminate the assignment of additional credit the instructor writes:

Column:	10	16
	PROF	NODATA, NOBONUS

Note: The operands may be placed in any order in the operand field.

e. Specification of messages

The instructor may send a message to the student whenever an incorrect answer is obtained and at the end of a grading run. These grader messages are defined through the use of a macro-instruction called IMSG. The instructor may define as many as six messages in the control section. The labels for the messages are IMESS1 through IMESS5 for intermediate answer messages and IMESSF for a final message. Messages can be up to 132 characters in length.

EXAMPLE:

Column:	1	10	16
	IMESS4	IMSG	'message to student'

Note: This message would be printed if the student's fourth intermediate answer were incorrect.

The order in which these items must be defined in the control section is shown in the example in Appendix E. The instructor specifies these parameters in a control section with labels which indicate the particular problem. GINITO indicates the first problem

whereas GINIT11 indicates the twelfth problem. The control section can be assembled and an object deck obtained using the System/360 ASMAD procedure. The control section may then be linked to the SOS system by the linkage editor.

D. FEATURES AND LIMITATIONS OF THE GRADER

A comparison of the characteristics of this grader with the general properties of grading systems listed in the previous chapter is now presented. The SOS grader accomplishes the basic objective of determining whether an answer is correct or not and assigning a grade.

The student involvement with the grading system itself has been kept to a minimum. Very little is required of the student to indicate the problem to be graded and correct answers.

The student can be assisted in his programming by messages from the instructor. This feature allows up to six messages from the instructor to the student. These messages are predefined to fit the circumstances of the problem solution.

Through the use of a general communications section, ease of specifying the answers and messages for various problems is made possible. Macro-instructions for use with the answer control section allow the instructor to provide grader data with minimum effort.

A feature of the grader is its flexibility in grading as many as six answers for one problem and its ability to grade twelve different problems during one batch run of student jobs. Students may work on several projects at the same time because of this capability.

The substitution of correct intermediate answers allows independent grading of the subsections of the student's program. By evaluating each section of a program separately, a more representative grade can be assigned.

The only feature of the grader which examines the quality of a student's program is the examination of the number of instructions executed. A better measurement of the efficiency of the program would be the execution time of the student program. The grader does not evaluate storage utilization or the number of runs submitted prior to grading. Inherent characteristics of the SOS system can be used by the instructor to assign a grade based on these parameters. These characteristics are discussed in Appendix A.

The SOS system uses word-oriented storage. Because of this, the grader is designed to evaluate fullword answers. A minor limitation is the fixed credit structure of the grader. However, this can be modified to suit a particular instructor's desires. Since the grader is designed for use in beginning programming classes, these limitations do not affect the capability to grade the type of program expected in introductory class work.

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

A grading system for use with the Student Operating System in beginning assembly language classes has been implemented at the U.S. Naval Postgraduate School. This system will significantly aid the instructor who must grade student exercises in classes with a large number of students.

This type of grader will provide the capability to assign more exercises during the academic quarter. Programming instruction can emphasize various aspects of the language in each assignment, thus allowing the student to gain more experience in assembly language programming.

Since the evaluation of a programming exercise is inevitably repetitive, the concept of writing a grader program which would be able to check other programs is both practical and useful. Documentation of previous grading systems has emphasized the implementation of the grading program. This investigation has examined the overall concept of a grading system. Characteristics of a grader have been evaluated. Several of these characteristics which measure the quality of a program would be extremely difficult to implement at the present time. Despite these complexities, it is anticipated that grading systems of the future will concentrate on the evaluation of the quality of a program.

A significant quality of grading systems in general is the fact that the machine may be more objective in grading than the human, because of its notable lack of prejudice and its inability to become bored [3].

Grading systems generate a paradoxical situation. A system which is designed to minimize student involvement with the grader itself, invariably requires more effort on the part of the instructor. A grader which not only checks answers but also evaluates the quality of the program may require considerable additional labor by the instructor. Because of this situation, tradeoff between student requirements and instructor efforts must be fully evaluated prior to the design of a grading system.

The grading system should allow the instructor flexibility in generating class assignments by minimizing programming work on his part with respect to the grader. It should be recognized however, that once the initial effort is expended in generating the problem, it can be utilized many times.

When a grading system is to be programmed, it behooves its designer to thoroughly examine the system with which it will interact. The communication capabilities and limitations of the system must be evaluated. The system parameters such as execution time, the number of instructions executed, and storage limitations should be examined before the initial design of a grading system. In this manner, various existing facilities may be used in the grading system.

B. RECOMMENDATIONS

It is strongly recommended that the grading system be used at the Naval Postgraduate School. It represents a powerful aid to assembly language programming instruction.

Recommendations for augmentation of the grader are:

(a) Inclusion of the standard data option of the Student Operating System to be used in conjunction with the grader.

(b) Revision of the Student Operating System statistics keeper so that it can produce a grade book based on results obtained from the grader.

(c) Modification of the grader by replacing student supervisor calls to the grader with simple Student Operating System macro-instructions.

(d) Provide the Student Operating System with a meaningful timer and revise the grader to utilize execution time as a measure of the efficiency of student programs.

(e) Assignment of these recommendations to students as suitable term projects in an assembly language programming class.

Very little work has been done in developing effective techniques to automatically evaluate the quality of a program. It is recommended that further investigation of both the theoretical and the practical phases of this area of grading be conducted.

APPENDIX A

The Student Operating System

A. PROGRAMMING INSTRUCTION AT THE POSTGRADUATE SCHOOL

Students taking a beginning Computer Science course at the Naval Postgraduate School have ample opportunity for actual machine experience. This is not the case for some universities where cost considerations take precedence and research work usually justifies the existence of the university computer center.

To add credence to the idea of efficiently instructing programming courses, an indirect measure of the amount of programming instruction at the Postgraduate School was obtained. This measure is the number of WATFOR jobs processed. WATFOR is an acronym for the Waterloo FORTRAN compiler. This system has been used extensively for introductory FORTRAN programming classes since October, 1967.

Table II presents the Naval Postgraduate School Computer Facility usage data. Approximately forty percent of the total number of computing jobs submitted in the past eighteen months have been WATFOR jobs. This is an indication that a primary use of the Computer Facility is programming instruction. Many of the jobs listed in the STUDENT category of Table II are also programming instruction class work. Faculty programming jobs comprise 8.75 percent of the total and are assumed to be primarily associated with research studies. This evidence indirectly substantiates the fact that a major use of the Computer Facility is programming instruction.

TABLE II

U. S. NAVAL POSTGRADUATE SCHOOL COMPUTER FACILITY USAGE DATA

(NUMBER OF COMPUTING JOBS SUBMITTED)

MONTH/ YEAR	SOURCE OF INPUT					
	STUDENT	FACULTY	COMPUTER FACILITY	MISCELLANEOUS	WATFOR	TOTAL
MAR 69	8,395	2,793	351	300	5,083	16,922
FEB 69	11,455	1,121	534	260	4,142	13,370
JAN 69	5,340	960	550	216	2,968	7,066
DEC 68	3,763	461	181	60	1,977	4,465
NOV 68	14,808	775	371	62	7,940	16,016
OCT 68	12,032	969	442	167	7,227	13,610
SEP 68	10,427	673	461	178	2,866	11,739
AUG 68	14,420	838	538	206	8,031	16,002
JUL 68	12,179	796	512	194	8,242	13,681
JUN 68	4,978	826	502	174	2,210	6,480
MAY 68	17,475	771	566	193	8,702	19,005
APR 68	13,384	736	493	96	6,111	14,709
MAR 68	9,777	853	698	116	4,848	11,444
FEB 68	8,239	1,175	496	54	4,013	9,964
JAN 68	4,519	1,066	483	65	1,718	6,133
DEC 67	4,068	740	351	61	1,489	5,220
NOV 67	9,582	1,306	398	56	2,577	11,342
OCT 67	7,366	1,183	348	77	2,065	8,974

B. THE ORIGINAL STUDENT OPERATING SYSTEM

A system which can be used for assembly language instruction is the Brown University Student Operating System [10]. This student operating system provides an assembler which produces code for a simplified machine, an interpreter which simulates the simplified machine and a control program for storage manipulation, editing of the student's program and statistics gathering. Wile [10] documents the capability of this system to provide assembly language teaching and processing at costs comparable to other university algebraic compilers used for student programming instruction.

The Student Operating System (SOS) offers several advantages to the instructor who wishes to utilize the IBM System/360 in an assembly language class. The feature of changing basic parameter settings at system generation time or taking default options allows the instructor to change the capabilities of the entire system for particular programming projects. All of the system is in core at all times and set up for each student job requires less than twenty machine instructions. SOS utilizes a simplified job control language, thus practically eliminating the difficult task of explaining the use of and reason for the complex, but extremely powerful, general purpose JCL required by the System/360 Operating System. Student jobs are cataloged on a disk; editing is done directly on the cataloged programs, thus introducing the student to the concept of text editing and library maintenance as well as reducing the volume of cards which the computer is required to read.

The SOS machine and assembly languages were designed to eliminate some of the more difficult programming concepts inherent in the

System/360 machine and assembly languages [10]. However, the SOS processing languages provide a compatible introduction to the related System/360 languages as the SOS assembly language mnemonics are, in general, a subset of the System/360 mnemonics. Arithmetic and logical operations are performed in a similar manner to System/360 operations. Data is in two's complement form and operations are performed in registers producing results and interruptions similar to 360 instructions.

The instructor may view the elimination of base/displacement addressability considerations, variable length instruction and data formats, floating point, packed decimal and character format conversions, relocatability, control and dummy section capabilities, and condition code testing from the assembly language repertoire as being too restrictive for the proper introduction of assembly language programming to the student. However, it is the opinion of this author that provisions included in the SOS system for indexing, indirect addressing, register addressing, overflow testing, and all arithmetic and logical operations are adequate to accomplish the objective of assembly language instruction at this school. A reasonable statement of this objective is to provide the student with an understanding of the structure and organization of a computing machine. It is not the intent of this study to document in detail the Student Operating System and its language. Specifications for the system and a description of its use may be found in Ref. 11.

C. IMPLEMENTATION OF THE SOS SYSTEM

The SOS System was obtained from Brown University in October, 1968. Documentation of the system received was negligible. System

generation was accomplished by issuing a single macro instruction (SOS); all program segments were included as lower level macro calls. This version was unsuitable for practical use in implementing a grader program within the system or for making changes to the system in general. The "SOS" macro included system default parameters. Required global set symbols were included in this macro. The main section of the "SOS" macro was the SOS processor common area. This section called the eight major macros in the system. The system assembled as the single macro-instruction "SOS" with this structure.

The IBM System/360 assembly language includes the COPY instruction. This instruction obtains source language coding from a library and includes it in the program currently being assembled. The assembler inserts the requested coding immediately after the COPY instruction is encountered. This instruction was utilized to provide a system structure which allowed manipulation of each of the eight major sections separately. The macro structure of each of the previously mentioned macros was removed and all global set symbols were placed in a section named SOSGBL which was then used as copy text in the SOS macro. Keyword parameters in the SOS macro were modified to make them compatible with the revised structure. In addition, the SOS macro was shortened to include only the necessary set symbol instructions. The original SOS processor common area was modified to call in the eight major program sections of the system by the use of COPY statements instead of the previous macro-expansion method. This section of coding was renamed SOSMAIN. The modified version of the system was generated on 7 February 1969.

D. SOS CHARACTERISTICS FOR STUDENT EVALUATION

Several inherent characteristics of the SOS system interrelate with the concept of grading a student's programming work. In describing the operation of a grader in use at Stanford University, Forsythe and Wirth [2] state that it is inappropriate to grade beginning students on the execution time of their programs or to evaluate the amount of storage used by their programs. Certainly the novice programmer should not be severely restricted in his programming efforts by these parameters. However, he should be made aware of their existence and thus eliminate the breeding of poor programming habits at the outset of instruction. By using the capability of varying system parameters in the SOS system the instructor may impose restrictions which will require the student to consider such aspects of programming as storage utilization, the number of instructions executed, register usage, and the number of program runs to achieve a solution.

The capability of imposing restraints on the student is essentially a method of evaluating his ability to program within the restricted environment. It is not an absolute grading mechanism, but rather a circuitous method of examining his programming ability. The features of the SOS system which provide this capability are discussed in the subsequent paragraphs of this section.

The highest configuration of SOS core the instructor wishes his students to use can be set by the keyword parameter MAXCNF = (1/2/..../8) at system generation time or by the bookkeeping parameter CONF = (1/2/..../8) at execution time. The student core size is then set to the value specified times 512 words. An instructor could generate a problem whose solution required approximately 1000 words

of storage and then specify in the assignment that the bookkeeping parameter CONF = 2 must be used on the student's job control card. If the student solved the problem within a 1024 storage environment, he could be given additional credit. The poorer student could still achieve the solution but would use the default parameter of 8 (4096 words). The actual configuration used is printed on the first page of the student's output by the bookkeeper. The ability to vary student core size is beneficial to the student because he can be made aware of real-world machine limitations at the start of his programming experience.

One means of student evaluation is to limit the number of programming runs that a student may submit prior to final grading of a project. This method insures that the student will carefully think out each step of his program and thoroughly review each instruction before submitting it for execution.

The SOS system provides the capability to record the number of runs a student submits for execution. If the system is used properly, students will be required to catalogue their programs in the student job library on a disk. All changes to the program are then made through the use of the SOS editing feature. On the first page of the student's output under the data card listing a message is printed "NEXT RUN IS NO.----." This provides the instructor with a means of evaluating the ability of the student to achieve a solution to the problem in as few runs as possible. Again this characteristic of the system does not pass an absolute judgement on a student's solution to a programming project. It does however, provide a better means of evaluating the overall potential of the student.

A third built-in grading characteristic of the SOS system is the capability to vary the maximum number of allowable executable instructions. The primary purpose of this feature is to prevent infinite looping. The maximum number of allowed executable instructions may be set at system generation time by the keyword parameter SOSMAX = (, , maximum number of instructions) or at execution time by the student through the use of the bookkeeping parameter INSC = (decimal number) on the student's job card. This parameter is evaluated directly by the Grader and is discussed in Chapter III.

In summary, the major capabilities of the SOS system with respect to student program evaluation are storage utilization flexibility, the capability to record every attempt at an exercise, and variability of the maximum allowable number of executable instructions. Imaginative and prudent application of these features of the SOS system by an instructor should motivate the student toward good programming practices and at the same time allow a broader evaluation of the true capability of the student.

APPENDIX B

The Grader Program

The Grader Program is shown in assembled form. The Grader Program is assembled with the Student Operating System when the keyword parameter GRADER=YES is chosen at system generation time.

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                 0124 ********************************************************************************
                                 0125 ********************************************************************************
                                 0126 ********************************************************************************
                                 0127 ***                                                                          ***
                                 0128 ***                                                                          ***
                                 0129 ***                                                                          ***
                                 0130 ***    GGGGGGG    RRRRRRR     AAAAAA    DDDDDD     EEEEEEEEE   RRRRRRR   ***
                                 0131 ***   GGGGGGGGG   RRRRRRRR   AAAAAAAA   DDDDDDD    EEEEEEEEE   RRRRRRRR  ***
                                 0132 ***   GG     GG   RR     RR  AA     AA  DD    DD   EE          RR     RR ***
                                 0133 ***   GG          RR     RR  AA     AA  DD     DD  EE          RR     RR ***
                                 0134 ***   GG          RR     RR  AA     AA  DD     DD  EE          RR     RR ***
                                 0135 ***   GG          RRRRRRRR  AAAAAAAAA  DD     DD  EEEEE       RRRRRRR  ***
                                 0136 ***   GG   GGGGG  RRRRRRR   AAAAAAAAA  DD     DD  EEEEE       RRRRRR   ***
                                 0137 ***   GG     GG   RRRR      AA     AA  DD     DD  EE          RR RR    ***
                                 0138 ***   GG     GG   RR RR     AA     AA  DD     DD  EE          RR  RR   ***
                                 0139 ***   GG     GG   RR  RR    AA     AA  DD    DD   EE          RR   RR  ***
                                 0140 ***   GGGGGGGGG   RR   RR   AA     AA  DDDDDDD    EEEEEEEEE   RR    RR ***
                                 0141 ***    GGGGGGG    RR    RR  AA     AA  DDDDDD     EEEEEEEEE   RR     RR ***
                                 0142 ***                                                                          ***
                                 0143 ***                                                                          ***
                                 0144 ***                                                                          ***
                                 0145 ********************************************************************************
                                 0146 ********************************************************************************
                                 0147 ********************************************************************************

                                 0148 *        ************************************************************        *
                                 0149 *        ************************************************************        *
                                 0150 *        **********                                          **********        *
                                 0151 *        **********                                          **********        *
                                 0152 *        **********   SOS CONTROL PROGRAM ENTRY AND RETURN   **********        *
                                 0153 *        **********                                          **********        *
                                 0154 *        **********                                          **********        *
                                 0155 *        ************************************************************        *
                                 0156 *        ************************************************************        *
                                 0157 *        ************************************************************        *

016D58                           0158 GRADER   CSECT
                                 0159          ENTRY ISVC3             PROVIDE CROSSREF BETWEEN GRADER & CALLS
000038                           0160          USING INTERRPT,3        TELL ASSEMBLER WHAT BASE TO USE.  NOTE:
000000                           0161          USING COMMON,4          SYMB ADDR OF 1ST OPERAND SERVES AS THE
002100                           0162          USING SOSCSECT,13       ORIGIN POINT THE ASSEMBLER USES FOR CAL
016D58                           0163          USING GRADER,15         CULATING DISPLACEMENTS FOR SYMB. NAMES
016D58 47EC F10E         16E66   0164          B     GHEADR            SKIP HEADING DEFINITION
                                 0165 GHEAD    GMESS L132'THIS IS AN SOS GRADER RUN -- GOOD LUCK ',C=3
016D5C                           0166+         ORG   *+2
016D5E                           0167+GHEADM   DS    DCL132
016D5E                           0168+         ORG   *-2
016D5E 83                        0169+GHEAD    DC    AL1(131)
016D5F 03                        0170+         DC    XL1'3'
016D60 E3C8C5E240C5E240          0171+         DC    CL132'THIS IS AN SOS GRADER RUN -- GOOD LUCK '
016DE2                           0172+         ORG
016DE2 4040404040404040          0173          DC    CL132'   '        BLANK SECOND LINE OF HEADING
016DE4 50EC F12C         16E94   0174 GHEADR   ST    14,SAVE14         SAVE RETURN
016DE8 4110 F0C4         16D5C   0175          LA    1,GHEAD           ADDRESS OF HEADER FOR PRINTER
016DEC 18CE                      0176          LR    12,15             SAVE BASE REGISTER
016DEE 45E0 D158         02258   0177          BAL   14,SOSPRINT       PRINT HEADING
016DF2 18EC                      0178          LR    15,12             RESTORE BASE REGISTER
016DF4 1888                      0179          SR    8,8               ZERO ACCUMULATOR
016DF6 50B0 EA54         17740   0180          ST    8,SUM
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
016E7C 5830 F130              16F8  3192           L     3,=V(INTERRPT)   GET ADDRESS OF INTERRPT
016E80 07F3                         3193           BR    3                GO TO INTERPRETER FOR EXEC. OF 1ST INST

016E84                              3195 SAVE14    DS    F
016E88                              3196           LTORG
016E88 00000000                     3197                  =V(INTERRPT)
                                    3198           DROP  15
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                 8191 *        ************************************************************
                                 8192 *        ************************************************************
                                 8193 *        **********                                      **********
                                 8194 *        **********                                      **********
                                 8195 *        **********      GRADER INITIALIZATION SECTION   **********
                                 8196 *        **********                                      **********
                                 8197 *        **********                                      **********
                                 8198 *        ************************************************************
                                 8199 *        ************************************************************

016E8C 18CF                      8201 ISVC3    LR    12,15
016E8E                           8202          USING ISVC3,12          ESTABLISH ADDRESSABILITY
016E8E 50EC C234          170C0  8203          ST    14,SAVE214        SAVE RETURN
016E92 4155 00C1          00001  8204          LA    5,1(5)            INCREMENT CONTROL COUNTER TO POINT AT
                                 8205 *                                THE NEXT INSTR. THE GRADER ALWAYS
                                 8206 *                                WORKS WITH TWO STUDENT INSTRUCTIONS.
016E96 1815                      8207          LR    1,5               CHECK FOR ...
016E98 45E0 4C30          00030  8208          BAL   14,IWRAP          STUDENT CORE WRAPAROUND
016E9C 47CC 0000          00000  8209          NOP   0                 ONE WRAP IS OK
016EAC 47CC C0CC          00000  8210          NOP   0                 NORMAL
016EA4 1851                      8211          LR    5,1
016EA6 4820 404C          0004C  8212          LH    2,IEFFA           LOAD SVC CODE
016EAA 918C 4004          00004  8213          TM    GCB,X'80'         IS THE GRADER IN?
016EAE 478C CCE6          16F72  8214          BE    CHKSVC            IF NO, BRANCH AND CONTINUE STUDENT RUN
016EB2 914C 4004          00004  8215          TM    GCB,X'40'         IS GRADING IN PROCESS?
016EB6 4780 CC38          16EC4  8216          BZ    GINIT             IF NO, BRANCH TO INITIALIZE GRADER
016EBA 5890 C238          170C4  8217          L     9,R9SAVE          ADDRESS FOR ANSWER COMMUNICATIONS SECT
016EBE 58F0 C23C          170C8  8218          L     15,=A(GMAIN)      OTHERWISE GO TO THE MAIN GRADER
016EC2 07FF                      8219          BR    15
016EC4 4920 C24C          170D8  8220 GINIT     CH    2,=H'0'           IS THE SVC CODE ZERO?
016EC8 4770 C0BE          16F4A  8221          BNE   GERR1             IF NOT, STUDENT HAS NOT INITIALIZED
016ECC 1825                      8222          LR    2,5               CHECK PROBLEM NUMBER NEXT
016ECE 892C 0002          00002  8223          SLL   2,2               MULTIPLY SCS BY 4
016ED2 5822 DEF8          02FF8  8224          L     2,ICORE(2)        LOAD PROBLEM NUMBER
016ED6 4920 C24E          170D4  8225          CH    2,=H'11'          CAN'T BE MORE THAN 12 PROBS
016EDA 472C C106          16F92  8226          BH    GINERR            PRINT ERROR MSG AND ABEND
016EDE 5020 C230          170BC  8227          ST    2,PROBSAVE        SAVE FOR USE IN CALLING ROUTINE
016EE2 964C 4004          00004  8228 INPROC   CI    GCB,X'40'         SET THE GRADER "IN PROCESS"
016EE6 96FF 4005          CCCC5  8229          CI    GCB+1,X'FF'       SET INSTRUCTOR'S CONTROL BLOCK
016EEA 892C 0003          00003  8230          SLL   2,3               MULTIPLY BY 8 FOR INDEX USE
016EEE 4110 C250          170DC  8231          LA    1,=CL34'  **          INITIALIZATION BEGUN***'
016EF2 4122 C1D0          1705C  8232          LA    2,INITNM(2)       GET ADDRESS OF CORRECT PROB. DESIGNATOR
016EF6 D207 10C4 2000 00C04 00000 8233          MVC   4(8,1),0(2)       PUT DESIGNATOR IN THE MESSAGE
016EFC D2C1 10C0 C272 CCCCC 170FE 8234          MVC   0(2,1),=X'1F02'   MOVE IN LENGTH OF LINE-1 AND PRINT CTRL
016F02 58F0 C240          170CC  8235          L     15,=A(SOSPRINT)   NOW, PRINT IT
016F06 05EF                      8236          BALR  14,15
016F08 5820 C230          170BC  8237          L     2,PROBSAVE        GET PROBLEM NUMBER BACK
016F0C 892C 0002          00002  8238          SLL   2,2               MULT BY 4 AGAIN FOR INDEX USE
016F10 5892 C1A0          1702C  8239          L     9,GINITBLE(2)     LOAD ENTRY POINT
016F14 5090 C238          170C4  8240          ST    9,R9SAVE          SAVE ADDRESS
016F18 1299                      8241          LTR   9,9               IF NOT THERE, ERROR
016F1A 478C C106          16F92  8242          BE    GINERR
CCCCCC                           8243          USING GINITDS,9         ESTABLISH BASE FOR COMMUNICATIONS DSECT
016F1E D40C 4005 9021 CCC5 00021 8244          NC    GCB+1(1),CONTROL+1    SET INSTRUCTOR OPTIONS
                                 8245          GPRT  31,1C,02,'  **INITIALIZATION SUCCESSFUL**'
016F24 4110 C2E4          17170  8246+         LA    1,=CL31'  **INITIALIZATION SUCCESSFUL**'
016F28 D2C1 1000 C274 CCCCC 17100 8247+         MVC   0(2,1),=X'1C02'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
016E2E 58E0 C240              17CCC  8248+          L     15,=A(SOSPRINT)
016E32 05EF                         8249+          BALR  14,15
                                    8250           GPRT  4,01,02,'    '
016E34 411C C244              170DC  8251+          LA    1,=CL4'    '
016E38 D201 1000 C276 CCOCC 171C2  8252+          MVC   0(2,1),=X'0102'
016E3E 58E0 C240              17CCC  8253+          L     15,=A(SOSPRINT)
016E42 05EF                         8254+          BALR  14,15
016E44 58EC C234              17CCC  8255 ISVCOUT   L     14,SAVE214        LOAD RETURN
016E48 07FE                         8256           BR    14                CONTINUE STUDENT PROGRAM

                                    8258 *         ************************************************************
                                    8259 *         **********                                      **********
                                    8260 *         **********  INITIALIZATION SECTION ERROR MESSAGES  **********
                                    8261 *         **********                                      **********
                                    8262 *         ************************************************************

016E4A 4110 C11A              16EA6  8264 GERR1    LA    1,GMESS1          LOAD ADDRESS OF GRADER ERROR MSG 1
016E4E 58E0 C240              17CCC  8265          L     15,=A(SOSPRINT)
016E52 05EF                         8266          BALR  14,15
                                    8267 ABEND    GPRT  18,0E,02,'  **ABNORMAL EOJ**'
016E54 411C C278              17104  8268+ABEND    LA    1,=CL18'  **ABNORMAL EOJ**'
016E58 D201 1000 C28A CCOOO 17116  8269+          MVC   0(2,1),=X'0E02'
016E5E 58E0 C240              17CCC  8270+          L     15,=A(SOSPRINT)
016E62 05EF                         8271+          BALR  14,15
016E64 943F 4004        CC004       8272           NI    GCR,X'3F'         PUT THE GRADER OUT
016E68 58E0 C248              170D4  8273          L     14,=A(SAVE14)     RETURN TO SOS CONTROL PROGRAM
016E6C 58EE 0000              CCOCC  8274          L     14,0(14)
016E70 07FE                         8275          BR    14
016E72 4920 C28C              17118  8276 CHKSVC   CH    2,=H'6'           IS SVC A VALID CODE?
016E76 4720 CCE2              16E7E  8277          BH    ERR1              IF NO, ERROR
016E7A 47EC CCB8              16E44  8278          B     ISVCOUT           OTHERWISE CONTINUE FOR NON-GRADING RUN
                                    8279 ERR1     GPRT  17,0E,02,'  **SVC UNKNOWN**'
016E7E 411C C303              1718E  8280+ERR1     LA    1,=CL17'  **SVC UNKNOWN**'
016E82 D201 1CO0 C28E CCOOC 1711A  8281+          MVC   0(2,1),=X'0E02'
016E88 58E0 C240              17CCC  8282+          L     15,=A(SOSPRINT)
016E8C 05EF                         8283+          BALR  14,15
016E8E 47EC CCC8              16E54  8284          B     ABEND             THAT'S ALL FOLKS
                                    8285 GINERR   GPRT  82,4E,02,'  **INITIALIZER ROUTINE SPECIFIED IS NOT AVAILX
                                    ABLE. CORRECT YOUR PROBLEM NUMBER.**'
                                    8286+GINERR   LA    1,=CL82'  **INITIALIZER ROUTINE SPECIFIED IS NOT AVAILABX
                                    LE. CORRECT YOUR PROBLEM NUMBER.**'
016E92 411C C29O              1711C  8287+          MVC   0(2,1),=X'4E02'
016E96 D201 1000 C2E2 CCCOC 1716E  8288+          L     15,=A(SOSPRINT)
016E9C 58EC C240              17CCC  8289+          BALR  14,15
016EA0 05EF                         8290          B     ABEND             BACK TO THE DRAWING BOARD
016EA2 47EC CCC8              16E54  8291 GMESS1    GMESS L132'**STUDENT HAS NOT INITIALIZED. CURRENT SVC NOT ACCEX
                                    PTABLE**',C=2
016EA8                              8292+          ORG   *+2
016EA8                              8293+GMESS1M   DS    0CL132
016EA6                              8294+          ORG   *-2
016EA4 83                           8295+GMESS1    DC    AL1(131)
016EA7 C2                           8296+          DC    XL1'2'
                                    8297+          DC    CL132'**STUDENT HAS NOT INITIALIZED. CURRENT SVC NOT ACCX
016EA8 5C5CE2E3F4C4C5D5
                                    EPTABLE**'
017020                              8298+          ORG

                                    8300           VADD  MGIN=(S,S,S,S,S,S,S,S,S,S,S,S)
017020 COOOOOOO               8301+GINITBLE DC    V(GINIT0) ENTRY POINT OF FIRST PROB INITIALIZER
017030 COOCCOOO               8302+          DC    V(GINIT1) DEFINE ENTRY POINT
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
017034 C00CC0C0                   8303+          DC    V(GINIT2) DEFINE ENTRY POINT
017038 C00C0000                   8304+          DC    V(GINIT3) DEFINE ENTRY POINT
01703C C00CC0C0                   8305+          DC    V(GINIT4) DEFINE ENTRY POINT
017040 C00C0C00                   8306+          DC    V(GINIT5) DEFINE ENTRY POINT
017044 C00CC000                   8307+          DC    V(GINIT6) DEFINE ENTRY POINT
017048 C00C0000                   8308+          DC    V(GINIT7) DEFINE ENTRY POINT
01704C C00C0000                   8309+          DC    V(GINIT8) DEFINE ENTRY POINT
017050 C00C0000                   8310+          DC    V(GINIT9) DEFINE ENTRY POINT
017054 C00CC00C                   8311+          DC    V(GINIT10) DEFINE ENTRY POINT
017058 C00C0000                   8312+          DC    V(GINIT11) DEFINE ENTRY POINT

01705C 07D9D6C24B40F14C           8314 INITNM    DC    CL8'PROB. 1'
017064 07D9D6C24B40F24C           8315           DC    CL8'PROB. 2'
01706C 07D9D6C24B40F34C           8316           DC    CL8'PROB. 3'
017074 07D9D6C24B40F44C           8317           DC    CL8'PROB. 4'
01707C 07D9D6C24B40F54C           8318           DC    CL8'PROB. 5'
017084 07D9D6C24B40F64C           8319           DC    CL8'PROB. 6'
01708C 07D9D6C24B40F74C           8320           DC    CL8'PROB. 7'
017094 07D9D6C24B40F84C           8321           DC    CL8'PROB. 8'
01709C 07D9D6C24B40F94C           8322           DC    CL8'PROB. 9'
0170A4 07D9D6C240F1F04C           8323           DC    CL8'PROB 10'
0170AC 07D9D6C240F1F14C           8324           DC    CL8'PROB 11'
0170B4 07D9D6C240F1F24C           8325           DC    CL8'PROB 12'
0170BC                            8326 PROBSAVE  DS    F
0170C0                            8327 SAVE214   DS    F
0170C4                            8328 RBSAVE    DS    F
0170C8                            8329           LTORG
0170C8 C0017180                   8330                 =A(GMAIN)
0170CC C0002258                   8331                 =A(SQSPRINT)
0170D0 40404040                   8332                 =CL4'    '
0170D4 C0016FB4                   8333                 =A(SAVE14)
0170D8 C00C                       8334                 =H'0'
0170DA C00B                       8335                 =H'11'
0170DC 40405C5C4040404C           8336                 =CL34'  **         INITIALIZATION BEGUN***'
0170FE 1F02                       8337                 =X'1F02'
01710C 1C02                       8338                 =X'1C02'
017102 0102                       8339                 =X'0102'
017104 40405C5CC1C2D5D6           8340                 =CL18'  **ABNORMAL EOJ**'
017116 0F02                       8341                 =X'0F02'
017118 0006                       8342                 =H'6'
01711A 0F02                       8343                 =X'0F02'
                                  8344                 =CL82'  **INITIALIZER ROUTINE SPECIFIED IS NOT AVAILABLEX
                                                       . CORRECT YOUR PROBLEM NUMBER.**'
01711C 40405C5CC9D5C9E3
017124 4F02                       8345                 =X'4F02'
017170 40405C5CC9D5C9E3           8346                 =CL31'  **INITIALIZATION SUCCESSFUL**'
017188 40405C5CE2E5C34C           8347                 =CL17'  **SVC UNKNOWN**'
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                  8350 *        ************************************************************************
                                  8351 *        ************************************************************************
                                  8352 *        **********                                                  **********
                                  8353 *        **********                                                  **********
                                  8354 *        **********            MAIN GRADER SECTION                   **********
                                  8355 *        **********                                                  **********
                                  8356 *        **********                                                  **********
                                  8357 *        ************************************************************************
                                  8358 *        ************************************************************************

017180 18CF                       8360 GMAIN    LR    12,15
017180                            8361          USING GMAIN,12          ESTABLISH ADDRESSING
017182 4920 C7CC          17986   8362          CH    2,=H'6'           IS SVC CODE VALID? - FOR FINAL ANSWER?
017186 4720 C450          175EC   8363          BH    GERR3             ERROR
01718A 4780 C258          173E8   8364          BE    FINANS            IF SVC 3,6 - GRADE FINAL ANSWER
01718E 4920 C7CE          17986   8365          CH    2,=H'1'
017192 4740 C450          175EC   8366          BL    GERR3             INVALID SVC FOR ANSWER DESIGNATION
017196 8920 0002          00002   8367          SLL   2,2               MULTIPLY BY FOUR FOR INDEX
01719A 47F2 CC1A          171BA   8368          B     *(2)              CALL INT. ANSWER GRADING ROUTINE
01719E 47F0 C032          171D2   8369          B     INTANS1           BRANCH TO APPROPRIATE ROUTINE
0171C2 47F0 CCA0          17240   8370          B     INTANS2
0171C6 47F0 C10E          172AE   8371          B     INTANS3
0171CA 47F0 C170          17310   8372          B     INTANS4
0171CE 47F0 C1E8          173BA   8373          B     INTANS5

                                  8375 *        ************************************************************************
                                  8376 *        **********                                                  **********
                                  8377 *        **********   INTERMEDIATE ANSWER GRADING ROUTINES           **********
                                  8378 *        **********                                                  **********
                                  8379 *        ************************************************************************

0171D2 45E0 C432          175D2   8381 INTANS1  BAL   14,GETANS          OBTAIN ANSWER FROM STUDENT
0171D6 D503 9000 C614 C0000 177B4 8382          CLC   ANS1,STUANS       IS THE STUDENT'S ANSWER CORRECT?
0171DC 4770 CC54          171E4   8383          BNE   ERR2               IF WRONG, ERROR MSG AND SUBSTITUTE
                                  8384          GPRT  39,24,01,'  **FIRST INTERMEDIATE ANSWER CORRECT**'
0171E0 4110 C97E          17B1E   8385+         LA    1,=CL39'  **FIRST INTERMEDIATE ANSWER CORRECT**'
0171E4 D201 1000 C7DC CC00C 17970 8385+         MVC   0(2,1),=X'2401'
0171EA 58E0 C738          178D8   8387+         L     15,=A(SOSPRINT)
0171EE 05EF               8388+         BALR  14,15
0171F0 47F0 C400          175AC   8389          B     MACHGRD           BRANCH TO TOTAL CREDIT FOR THIS ANSWER

0171F4 9108 4005          00005   8391 ERR2     TM    GCB+1,X'08'       DOES INSTRUCTOR HAVE ANYTHING TO SAY?
0171F8 4780 CC6A          172CA   8392          BE    ERR21             IF NOT, SEE WHAT INSTRUCTOR WANTS NEXT
0171FC 4110 9022          00022   8393          LA    1,IMESS1          OBTAIN ADDRESS OF INSTRUCTOR'S MESSAGE
017200 1211               8394          LTR   1,1               IS MESSAGE DEFINED?
017202 4780 CC6A          172CA   8395          BE    ERR21             IF NOT, NOOP
017206 45E0 D158          022F8   8396          BAL   14,SOSPRINT       PRINT INSTRUCTOR'S MESSAGE TO STUDENT

01720A 9104 4005          00005   8398 ERR21    TM    GCB+1,X'04'       DOES INSTRUCTOR WANT DATA SUBSTITUTED?
01720E 4780 CC9C          1722C   8399          BE    NODATA1           IF NOT, BRANCH
                                  8400          GPRT  81,4E,01,'  **FIRST INTERMEDIATE ANSWER NOT CORRECT. DATX
                                                A SUBSTITUTED, ROUTINE CONTINUED.**'
                                  8401+         LA    1,=CL81'  **FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA X
                                                SUBSTITUTED, ROUTINE CONTINUED.**'
017212 4110 CC45          17B45     +
017216 D201 1000 C7D2 CC000 17972 8402+         MVC   0(2,1),=X'4E01'
01721C 58E0 C738          178D8   8403+         L     15,=A(SOSPRINT)
017220 05EF               8404+         BALR  14,15
017222 D203 6000 9000 CC000 CCCCC 8405          MVC   0(4,6),ANS1       GIVE THE STUDENT THE CORRECT ANSWER
017228 47F0 C3D6          17796   8406          B     OUT               LET'S GO SEE WHAT THE STUDENT IS DOING
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                  8408 NODATA1  GPRT  85,52,02,'  **FIRST INTERMEDIATE ANSWER NOT CORRECT. DATX
                                                      A NOT SUBSTITUTED. ROUTINE CONTINUED.**'
                                  8409+NODATA1  LA    1,=CL86'  **FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA X
01722C 4110 C9E6            17896     +               NOT SUBSTITUTED. ROUTINE CONTINUED.**'
01723C D201 1000 C7D4 CCCCC 17974 8410+           MVC   0(2,1),=X'5202'
017236 58EC C738            178D8 8411+           L     15,=A(SOSPRINT)
017238 05EF                       8412+           BALR  14,15
01723C 47F0 C3D6            17576 8413            B     OUT


017240 45EC C432            175D2 8415 INTANS2    BAL   14,GETANS         OBTAIN STUDENT'S ANSWER
017244 D503 9004 C614 00004 17784 8416            CLC   ANS2,STUANS       IS THE STUDENT'S ANSWER CORRECT?
01724A 4770 C0C2            17262 8417            BNE   ERR3              IF WRONG, ERROR MSG AND SUBSTITUTE
                                  8418            GPRT  40,25,01,'  **SECOND INTERMEDIATE ANSWER CORRECT**'
01724E 4110 C6E8            17888 8419+           LA    1,=CL40'  **SECOND INTERMEDIATE ANSWER CORRECT**'
017252 D201 1000 C7D6 CCCCC 17976 8420+           MVC   0(2,1),=X'2501'
017258 58EC C738            178D8 8421+           L     15,=A(SOSPRINT)
01725C 05EF                       8422+           BALR  14,15
01725E 47F0 C400            175AC 8423            B     MACHGRD           BRANCH TO TOTAL CREDIT FOR THIS ANSWER

017262 9108 4005      00005       8425 ERR3       TM    GCB+1,X'08'       DOES INSTRUCTOR HAVE ANYTHING TO SAY?
017266 4780 C0D8            17278 8426            BE    ERR31             IF NOT, SEE WHAT INSTRUCTOR WANTS NEXT
01726A 4110 90A8            00A8  8427            LA    1,IMESS2          OBTAIN ADDRESS OF INSTRUCTOR'S MESSAGE
01726E 1211                       8428            LTR   1,1               IS MESSAGE DEFINED?
017270 4780 C0D8            17278 8429            BE    ERR31             IF NOT, NOOP
017274 45E0 D158            02258 8430            BAL   14,SOSPRINT       PRINT INSTRUCTOR'S MESSAGE TO STUDENT

017278 9104 4005      00005       8432 ERR31      TM    GCB+1,X'04'       DOES INSTRUCTOR WANT DATA SUBSTITUTED?
01727C 4780 C0FA            1729A 8433            BE    NODATA2           IF NOT, BRANCH
                                  8434            GPRT  82,4E,01,'  **SECOND INTERMEDIATE ANSWER NOT CORRECT. DAX
                                                      TA SUBSTITUTED. ROUTINE CONTINUED.**'
                                  8435+           LA    1,=CL82'  **SECOND INTERMEDIATE ANSWER NOT CORRECT. DATAX
017280 4110 C7D8            17978     +               SUBSTITUTED. ROUTINE CONTINUED.**'
017284 D201 1000 C82A CCCCC 179CA 8436+           MVC   0(2,1),=X'4F01'
01728A 58EC C738            178D8 8437+           L     15,=A(SOSPRINT)
01728E 05EF                       8438+           BALR  14,15
017290 D203 6000 9004 0000C 00004 8439            MVC   0(4,6),ANS2       GIVE THE STUDENT THE CORRECT ANSWER
017296 47F0 C3D6            17576 8440            B     OUT               LET'S GO SEE WHAT THE STUDENT IS DOING

                                  8442 NODATA2    GPRT  86,53,02,'  **SECOND INTERMEDIATE ANSWER NOT CORRECT. DAX
                                                      TA NOT SUBSTITUTED. ROUTINE CONTINUED.**'
                                  8443+NODATA2    LA    1,=CL86'  **SECOND INTERMEDIATE ANSWER NOT CORRECT. DATAX
01729A 4110 C82C            179CC     +               NOT SUBSTITUTED. ROUTINE CONTINUED.**'
01729E D201 1000 C882 CCCCC 17A22 8444+           MVC   0(2,1),=X'5302'
0172A4 58EC C738            178D8 8445+           L     15,=A(SOSPRINT)
0172A8 05EF                       8446+           BALR  14,15
0172AA 47F0 C3D6            17576 8447            B     OUT


0172AE 45EC C432            175D2 8449 INTANS3    BAL   14,GETANS         OBTAIN THE STUDENT'S ANSWER
0172B2 D503 9008 C614 00008 17784 8450            CLC   ANS3,STUANS       IS THE STUDENT'S ANSWER CORRECT?
0172B8 4770 C130            172DC 8451            BNE   ERR4              IF WRONG, ERROR MSG AND SUBSTITUTE
                                  8452            GPRT  39,24,01,'  **THIRD INTERMEDIATE ANSWER CORRECT**'
0172BC 4110 CA48            17BEA 8453+           LA    1,=CL39'  **THIRD INTERMEDIATE ANSWER CORRECT**'
0172C0 D201 1000 C7D0 C0000 17970 8454+           MVC   0(2,1),=X'2401'
0172C6 58EC C738            178D8 8455+           L     15,=A(SOSPRINT)
0172CA 05EF                       8456+           BALR  14,15
0172CC 47F0 C400            175A0 8457            B     MACHGRD           BRANCH TO TOTAL CREDIT FOR THIS ANSWER

0172D0 9108 4005      00005       8459 ERR4       TM    GCB+1,X'08'       DOES INSTRUCTOR HAVE ANYTHING TO SAY?
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
017204 4780 C146             172E6 9460           BE    ERR41             IF NOT, SEE WHAT INSTRUCTOR WANTS NEXT
017208 4110 912E             0012E 9461           LA    1,IMESS3          OBTAIN ADDRESS OF INSTRUCTOR'S MESSAGE
01720C 1211                        9462           LTR   1,1               IS MESSAGE DEFINED?
01720E 4780 C146             172E6 9463           BE    ERR41             IF NOT, NOOP
017212 45E0 D158             02258 9464           BAL   14,SOSPRINT       PRINT INSTRUCTOR'S MESSAGE TO STUDENT

017216 9104 4005       00005       9466 ERR41     TM    GCB+1,X'04'       DOES INSTRUCTOR WANT DATA SUBSTITUTED?
01721A 4780 C168             1730E 9467           BE    NODATA3           IF NOT, BRANCH
                                   9468           GPRT  81,4E,01,'  **THIRD INTERMEDIATE ANSWER NOT CORRECT. DATX
                                                       A SUBSTITUED. ROUTINE CONTINUED.**'
                                   9469+          LA    1,=CL81'  **THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA X
017220 4110 CA72             17C12   +                  SUBSTITUTED. ROUTINE CONTINUED.**'
017224 D201 1000 C7D2 C0000 17972  9470+          MVC   0(2,1),=X'4E01'
01722A 58F0 C738             178D8 9471+          L     15,=A(SOSPRINT)
01722E 05EF                        9472+          BALR  14,15
017230 D203 6000 9009 C0000 00000  9473           MVC   0(4,6),ANS3       GIVE THE STUDENT THE CORRECT ANSWER
017306 47F0 C3D6             17576 9474           B     OUT               LET'S GO SEE WHAT THE STUDENT IS DOING

                                   9476 NODATA3   GPRT  85,52,02,'  **THIRD INTERMEDIATE ANSWER NOT CORRECT. DATX
                                                       A NOT SUBSTITUTED. ROUTINE CONTINUED.**'
                                   9477+NODATA3   LA    1,=CL85'  **THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA X
017308 4110 CAC3             17C63   +                  NOT SUBSTITUTED. ROUTINE CONTINUED.**'
01730C D201 1000 C7D4 C0000 17974  9478+          MVC   0(2,1),=X'5202'
017312 58F0 C738             178D8 9479+          L     15,=A(SOSPRINT)
017316 05EF                        9480+          BALR  14,15
017318 47F0 C3D6             17576 9481           B     OUT


01731C 45E0 C432             175D2 9483 INTANS4   BAL   14,GETANS         OBTAIN THE STUDENT'S ANSWER
017320 D503 900C C614 0000C 177B4  9484           CLC   ANS4,STUANS       IS THE STUDENT'S ANSWER CORRECT?
017326 4770 C19E             1733E 9485           BNE   ERR5               IF WRONG, ERROR MSG AND SUBSTITUTE
                                   9486           GPRT  40,25,01,'  **FOURTH INTERMEDIATE ANSWER CORRECT**'
01732A 4110 C710             178BC 9487+          LA    1,=CL40'  **FOURTH INTERMEDIATE ANSWER CORRECT**'
01732E D201 1000 C7D6 00000 17976  9488+          MVC   0(2,1),=X'2501'
017334 58F0 C738             178D8 9489+          L     15,=A(SOSPRINT)
017338 05EF                        9490+          BALR  14,15
01733A 47FC C400             175AC 9491           B     MACHGRD           BRANCH TO TOTAL CREDIT FOR THIS ANSWER

01733E 9108 4005       00005       9493 ERR5      TM    GCB+1,X'08'       DOES INSTRUCTOR HAVE ANYTHING TO SAY?
017342 4780 C1B4             17354 9494           BE    ERR51             IF NOT, SEE WHAT INSTRUCTOR WANTS NEXT
017346 4110 91B4             001B4 9495           LA    1,IMESS4          OBTAIN ADDRESS OF INSTRUCTOR'S MESSAGE
01734A 1211                        9496           LTR   1,1               IS MESSAGE DEFINED?
01734C 4780 C1B4             17354 9497           BE    ERR51             IF NOT, NOOP
017350 45E0 D158             02258 9498           BAL   14,SOSPRINT       PRINT INSTRUCTOR'S MESSAGE TO STUDENT

017354 9104 4005       00005       9500 ERR51     TM    GCB+1,X'04'       DOES INSTRUCTOR WANT DATA SUBSTITUTED?
017358 4780 C1D6             17376 9501           BE    NODATA4           IF NOT, BRANCH
                                   9502           GPRT  82,4E,01,'  **FOURTH INTERMEDIATE ANSWER NOT CORRECT. DAX
                                                       TA SUBSTITUTED. ROUTINE CONTINUED.**'
                                   9503+          LA    1,=CL82'  **FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATAX
01735C 4110 C8B4             17A24   +                  SUBSTITUTED. ROUTINE CONTINUED.**'
017360 D201 1000 C92A C0000 17ACA  9504+          MVC   0(2,1),=X'4E01'
017366 58F0 C738             178D8 9505+          L     15,=A(SOSPRINT)
01736A 05EF                        9506+          BALR  14,15
01736C D203 6000 900C 0C0C0 0C0CC  9507           MVC   0(4,6),ANS4       GIVE THE STUDENT THE CORRECT ANSWER
017372 47F0 C3D6             17576 9508           B     OUT               LET'S GO SEE WHAT THE STUDENT IS DOING

                                   9510 NODATA4   GPRT  86,53,02,'  **FOURTH INTERMEDIATE ANSWER NOT CORRECT. DAX
                                                       TA NOT SUBSTITUTED. ROUTINE CONTINUED.**'
                                   9511+NODATA4   LA    1,=CL86'  **FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATAX
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
017376 4110 C8D6             17A76      +                NOT SUBSTITUTED. ROUTINE CONTINUED.**'
017378 D201 1000 C882 CCCCC  17A22  8512+          MVC   0(2,1),=X'5302'
01737C 58F0 C738             17BD8  8513+          L     15,=A(SOSPRINT)
017380 05EF                         8514+          BALR  14,15
017382 47F0 C3D6             17576  8515           B     OUT

017386 45E0 C432             175D2  8517 INTANS5   BAL   14,GETANS        OBTAIN THE STUDENT'S ANSWER
01738A D503 9010 C614 CCC1C  17784  8518           CLC   ANS5,STUANS       IS THE STUDENT'S ANSWER CORRECT?
017390 4770 C2CC             173AC  8519           BNE   ERR6              IF WRONG, ERROR MSG AND SUBSTITUTE
                                    8520           GPRT  39,24,01,'  **FIFTH INTERMEDIATE ANSWER CORRECT**'
017394 4110 C818             17C98  8521+          LA    1,=CL39'  **FIFTH INTERMEDIATE ANSWER CORRECT**'
017398 D201 1000 C7D0 CC000  17970  8522+          MVC   0(2,1),=X'2401'
01739E 58F0 C738             17BD8  8523+          L     15,=A(SOSPRINT)
0173A2 05EF                         8524+          BALR  14,15
0173A4 47F0 C400             175AC  8525           B     MACHGRD           BRANCH TO TOTAL CREDIT FOR THIS ANSWER

0173A8 9108 4005       00005        8527 ERR6     TM    GCB+1,X'08'       DOES INSTRUCTOR HAVE ANYTHING TO SAY?
0173AC 4780 C222             173C2  8528           BE    ERR61             IF NOT, SEE WHAT INSTRUCTOR WANTS NEXT
0173B0 4110 923A             0023A  8529           LA    1,IMESS5          OBTAIN ADDRESS OF INSTRUCTOR'S MESSAGE
0173B4 1211                         8530           LTR   1,1               IS MESSAGE DEFINED?
0173B6 4780 C222             173C2  8531           BE    ERR61             IF NOT, NOOP
0173BA 45E0 D158             02258  8532           BAL   14,SOSPRINT       PRINT INSTRUCTOR'S MESSAGE TO STUDENT

0173BE 9104 4005       00005        8534 ERR61    TM    GCB+1,X'04'       DOES INSTRUCTOR WANT DATA SUBSTITUTED?
0173C2 4780 C244             173E4  8535           BE    NODATA5           IF NOT, BRANCH
                                    8536           GPRT  81,4E,01,'  **FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATX
                                                       A SUBSTITUTED. ROUTINE CONTINUED.**'
                                    8537+          LA    1,=CL81'  **FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATA X
0173C6 4110 C83E             17CDE      +                SUBSTITUTED. ROUTINE CONTINUED.**'
0173CA D201 1000 C7D2 CC000  17972  8538+          MVC   0(2,1),=X'4E01'
0173D0 58F0 C738             17BD8  8539+          L     15,=A(SOSPRINT)
0173D4 05EF                         8540+          BALR  14,15
0173D6 D203 6000 9010 CCCC0  CC01C  8541           MVC   0(4,6),ANS5       GIVE THE STUDENT THE CORRECT ANSWER
0173DC 47F0 C3D6             17576  8542           B     OUT               LET'S GO SEE WHAT THE STUDENT IS DOING

                                    8544 NODATA5  GPRT  85,52,02,'  **FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATX
                                                       A NOT SUBSTITUTED. ROUTINE CONTINUED.**'
                                    8545+NODATA5  LA    1,=CL85'  **FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATA X
0173E0 4110 C890             17D3C      +                NOT SUBSTITUTED. ROUTINE CONTINUED.**'
0173E4 D201 1000 C7D4 CCCCC  17974  8546+          MVC   0(2,1),=X'5202'
0173EA 58F0 C738             17BD8  8547+          L     15,=A(SOSPRINT)
0173EE 05EF                         8548+          BALR  14,15
0173F0 47F0 C3D6             17576  8549           B     OUT

                                    8551 *        ************************************************************
                                    8552 *        **********                                      **********
                                    8553 *        **********       FINAL ANSWER GRADING ROUTINE   **********
                                    8554 *        **********                                      **********
                                    8555 *        ************************************************************

0173F4 45E0 C432             175D2  8557 FINANS    BAL   14,GETANS
0173F8 D503 9014 C614 CCC14  17784  8558           CLC   ANSF,STUANS       IS THE STUDENT'S ANSWER CORRECT?
0173FE 4770 C2B4             17424  8559           BNE   ERR7
                                    8560           GPRT  26,17,02,'  **FINAL ANSWER CORRECT**'
017402 4110 C92C             17ACC  8561+          LA    1,=CL26'  **FINAL ANSWER CORRECT**'
017406 D201 1000 C946 CC000  17AF6  8562+          MVC   0(2,1),=X'1702'
01740C 58F0 C738             17BD8  8563+          L     15,=A(SOSPRINT)
017410 05EF                         8564+          BALR  14,15
017412 1BBB                         8565           SR    8,8               ZERO FOR FUN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
017418 4188 0032             00C32  8566           LA    8,50(8)          FIFTY POINTS FOR THE FINAL ANSWER
01741C 5080 C610             177BC  8567           ST    8,SUM1           SAVE IT FOR TOTAL LATER
017420 47EC C2E4             17434  8568           B     GRADEIT          NOW, LET'S TOTAL HIS GRADE
                                    8569 ERR7       GPRT  28,19,02,'  **FINAL ANSWER INCORRECT**'
017424 4110 C73C             178DC  8570+ERR7       LA    1,=CL28'  **FINAL ANSWER INCORRECT**'
017428 D201 1000 C94A C0000  17AFA  8571+           MVC   0(2,1),=X'1902'
01742E 58EC C738             178D8  8572+           L     15,=A(SOSPPINT)
017432 05EF                         8573+           BALR  14,15

                                    8575 *          ********************************************************************
                                    8576 *          **********                                              **********
                                    8577 *          **********        TOTAL MACHINE GRADE ROUTINE           **********
                                    8578 *          **********                                              **********
                                    8579 *          ********************************************************************

017434 58AC 9018             00018  8581 GRADEIT   L     10,NOINTANS      HOW MANY INTERMEDIATE ANSWERS?
017438 59A0 C758             178F8  8582           C     10,=F'5'
01743C 4720 C464             176C4  8583           BH    GERR4            CAN'T BE MORE THAN FIVE
017440 47BC C2FC             174FC  8584           BE    FINGRD           IF FIVE, JUST TOTAL THE GRADE
017444 59AC C75C             178FC  8585           C     10,=F'0'
017448 474C C464             176C4  8586           BL    GERR4            INDEX SCHEME COULD BE IN TROUBLE HERE
01744C 47BC C2FC             174FC  8587           BE    FINGRD           IF SO, JUST TOTAL THE GRADE
017450 89A0 0002             C0002  8588           SLL   10,2             MULTIPLY BY FOUR FOR INDEX
017454 47FA C2B4             17454  8589           B     *(10)            USE FOR INDEX
017458 47EC C2C8             174F8  8590           B     GRD1             BRANCH TO GET MACHINE GRADE
01745C 47EC C2D8             17508  8591           B     GRD2             BRANCH TO GET MACHINE GRADE
017460 47EC C2E8             17518  8592           B     GRD3             BRANCH TO GET MACHINE GRADE
017464 47EC C2F4             17524  8593           B     GRD4             BRANCH TO GET MACHINE GRADE
017468 58BC C60C             177AC  8594 GRD1       L     8,SUM            OBTAIN ACCUMULATED INT. ANSWER CREDIT
01746C 4188 0028             00028  8595           LA    8,40(8)          40 SILVER $ TO THE MAN IN THE BALCONY
017470 50BC C60C             177AC  8596           ST    8,SUM            SAVE IT FOR THE FINAL CREDIT
017474 47EC C2EC             174FC  8597           B     FINGRD           NOW FOR THE TOTAL MACHINE GRADE
017478 58B0 C60C             177AC  8598 GRD2       L     8,SUM            OBTAIN ACCUMULATED INT. ANSWER CREDIT
01747C 4188 001E             C001E  8599           LA    8,30(8)          THIRTY FREEBIES THIS TIME
017480 50BC C60C             177AC  8600           ST    8,SUM            SAVE IT FOR THE FINAL CREDIT
017484 47EC C2EC             174FC  8601           B     FINGRD           NOW FOR THE TOTAL MACHINE GRADE
017488 58BC C60C             177AC  8602 GRD3       L     8,SUM            OBTAIN ACCUMULATED INT. ANSWER CREDIT
01748C 4188 0014             C0014  8603           LA    8,20(8)          ADD TWENTY POINTS
017490 47EC C2EC             174FC  8604           B     FINGRD           NOW FOR THE TOTAL MACHINE GRADE
017494 58BC C60C             177AC  8605 GRD4       L     8,SUM            OBTAIN ACCUMULATED INT. ANSWER CREDIT
017498 4188 000A             C000A  8606           LA    8,10(8)          SUPPLEMENT TO FIFTY AGAIN

01749C 58B0 C60C             177AC  8608 FINGRD     L     8,SUM            GET BACK CORRECT ACCUMULATED CREDIT
0174A0 5870 C610             177BC  8609           L     7,SUM1           GET CREDIT FOR FINAL ANSWER IF ANY
0174A4 1187                         8610           AR    8,7              ADD'EM UP
0174A6 4EB0 C618             177B8  8611 TRANS1     CVD   8,WORKWORD       TRANSLATE AND PRINTOUT TOTAL MACH GRADE
0174AA F331 C620 C61F 177C0 177BE  8612           UNPK  CREDIT(4),WORKWORD+6(2)
0174B0 96F0 C623      177C3        8613           OI    CREDIT+3,X'F0'
0174B4 D203 C4AD 43AD 17640 003AD  8614           MVC   GR1(4),IBLANK
0174BA D202 C4AE C621 1764E 177C1  8615           MVC   GR1+1(3),CREDIT+1
0174C0 4110 C478             17618  8616           LA    1,GRM1
0174C4 45E0 D158             02258  8617           BAL   14,SOSPRINT

0174C8 9108 4005      00005        8619           TM    GCB+1,X'08'      DOES INSTRUCTOR HAVE A FINAL MESSAGE?
0174CC 47B0 C33E             174DE  8620           BE    FINOUT           IF NOT, FINISH UP
0174D0 4110 92C0             002C0  8621           LA    1,IMESSE         ADDRESS OF INSTRUCTOR MESSAGE
0174D4 1211                         8622           LTR   1,1              DID THE INSTRUCTOR FORGET THE MESSAGE?
0174D6 47B0 C33E             174DE  8623           BE    FINOUT           IF SO, NOOP--HE'S HUMAN TOO
0174DA 45E0 D158             02258  8624           BAL   14,SOSPRINT      PRINT THE FINAL MESSAGE
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   9626 *        ****************************************************************
                                   9627 *        **********                                          **********
                                   9628 *        **********         INSTRUCTIONS EXECUTED ROUTINE    **********
                                   9629 *        **********                                          **********
                                   9630 *        ****************************************************************

017ADE D293 C62E 43AD 177CE 0C3AD  9632 FINOUT   MVC   GEND,IBLANK
017AE4 D212 C62E C682 177CE 17EE2  9633          MVC   GEND(19),GHALT1  MOVE IN FIRST PART OF HALT MESSAGE
017AEA 5810 402C                   9634          L     1,IINSCC         CURRENT INSTR. COUNT FOR UNPACK ROUT.
017AEE 45EC D61C                   C271C         BAL   14,IPHUD         UNPACK IT
017AF2 1820                        9636          LR    2,0
017AF4 0622                        9637          BCTR  2,0              GET LENGTH
017AF6 4422 C6DC                   17B7C         EX    2,GMVC           MOVE IT
017AFA 5A2C C760                   179CC         A     2,=A(GEND+22)
017AFE D21A 2000 C6C5 00000 17B6E  9640          MVC   0(23,2),GHALT2   MOVE IN SECOND PART OF HALT MESSAGE
017B04 4110 C62C                   177CC         LA    1,GEND-2         GIVE PRINTER ADDRESS OF MESSAGE
017B08 45EC D158                   C2258         BAL   14,SOSPRINT      PRINT IT

017B0C 9102 4005       00005       9644          TM    GCB+1,X'02'      DOES INSTRUCTOR WANT BONUS CHECK?
017B10 4780 C3C8                   17B6E         BE    FINOUT1          IF NOT, BRANCH
017B14 D503 901C 402C 001C CCC2C   9646          CLC   NCX,IINSCC       STUDENT USE FEWER INSTRUCTIONS?
017B1A 474C C3BE                   17B5E         BL    GERR6            NO, BRANCH AND TELL HIM SO
                                   9648 GERR7    GPRT  127,70,02,'  **YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS X
                                                       THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY. ADDITIONAX
                                                       L CREDIT IS GIVEN BELOW.**'
                                   9649+GERR7    LA    1,=CL127'  **YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS THX
                                      +                      AN THE MEDIAN FOR PROJECTS IN THIS CATEGORY. ADDITIONAL X
                                      +                      CREDIT IS GIVEN BELOW.**'
017B1E 4110 CBE5                   17D85         LA    1,=CL127'  **YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS THX
017B22 D201 1C00 C94A CC000 17AEA  9650+         MVC   0(2,1),=X'7C02'
017B28 58EC C738                   178D8         L     15,=A(SOSPRINT)
017B2C 05EF                        9652+         BALR  14,15
017B2E 41B9 0C0A                   00C0A         LA    9,10(8)
017B32 4E9C C618                   177B8         CVD   8,WORKWORD       TRANSLATE AND PRINTOUT EXTRA CREDIT
017B36 F331 C628 C61E 177C8 177BE  9655          UNPK  XCREDIT(4),WORKWORD+6(2)
017B3C 96F0 C628                   177C8         OI    XCREDIT+3,X'F0'
017B40 D203 C5C8 43AD 177B8 0C3AD  9657          MVC   GR3(4),IBLANK
017B46 D202 C5CC C629 1776C 177C8  9658          MVC   GR3+1(3),XCREDIT+1
017B4C 4110 C584                   17724         LA    1,GRM3
017B50 45EC D158                   C2258         BAL   14,SOSPRINT
017B54 47F0 C3C8                   17B6E         B     FINOUT1
                                   9662 GERR6    GPRT  92,59,02,'  **YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THX
                                                       AN THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**'
                                   9663+GERR6    LA    1,=CL92'   **YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THANX
                                      +                       THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**'
017B58 4110 C764                   17904         LA    1,=CL92'   **YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THX
017B5C D201 1C00 C94C CCC0C 17AFC  9664+         MVC   0(2,1),=X'5C02'
017B62 58EC C738                   178D8         L     15,=A(SOSPRINT)
017B66 05EF                        9666+         BALR  14,15

017B68 58EC C7C0                   17960         L     14,=A(SAVE14)    GET ADDRESS TO RETURN TO BOOKEEPER
017B6C 58EE 0000                   CCCCC         L     14,0(14)
017B70 943E 40C4       00C4        9670          NI    GCB,X'3F'        PUT THE GRADER OUT
017B74 07EE                        9671          BR    14               RETURN TO SOS CONTROL VIA BOOKKEEPER


                                   9674 *        ****************************************************************
                                   9675 *        **********                                          **********
                                   9676 *        **********         RETURN TO INTERPRETER ROUTINE    **********
                                   9677 *        **********                                          **********
                                   9678 *        ****************************************************************
```

```
   LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     8680 OUT      GPRT  4,01,02,'    '
017574 4110 C7C4                 17964 8681+OUT     LA    1,=CL4'    '
017578 D201 1000 C94E CC000      17AFE 8682+        MVC   0(2,1),=X'0102'
01757E 58F0 C738                 17BDE 8683+        L     15,=A(SOSPRINT)
017582 05EF                            8684+        BALR  14,15
                                     8685           GPRT  4,01,02,'    '
017584 4110 C7C4                 17964 8686+        LA    1,=CL4'    '
017588 D201 1000 C94E CC000      17AFE 8687+        MVC   0(2,1),=X'0102'
01758E 58F0 C738                 17BDE 8688+        L     15,=A(SOSPRINT)
017592 05EF                            8689+        BALR  14,15
017594 58F0 C7C8                 17968 8690         L     14,=A(SAVE214)   GET ADDRESS TO RETURN TO INTERPRETER
017598 58EE 0000                 CCCCC 8691         L     14,0(14)
01759C 07FE                            8692         BR    14               RETURN TO INTERPRETER FOR FURTHER EXEC

                                     8694 *         ********************************************************************
                                     8695 *         **********                                              **********
                                     8696 *         **********    INTERMEDIATE ANSWER CREDIT ROUTINE        **********
                                     8697 *         **********                                              **********
                                     8698 *         ********************************************************************

0175A0 588C C60C                 177AC 8700 MACHGRD L     8,SUM            OBTAIN ACCUMULATED CREDIT, IF ANY
0175A4 4188 000A                 0000A 8701         LA    8,10(8)          ADD 10 POINTS FOR CORRECT INTERMEDIATE
0175A8 508C C60C                 177AC 8702         ST    8,SUM            STORE IT IN THE ACCUMULATOR  (SUM)
0175AC 4E8C C619                 177B8 8703 TRANS3  CVD   8,WORKWORD       TRANSLATE AND PRINTOUT ACCUM INT GRADE
0175B0 F331 C624 C61E 177C4 177BE 8704         UNPK  ICREDIT(4),WORKWORD+6(2)
0175B6 95F0 C627                 177C7 8705         CI    ICREDIT+3,X'F0'
0175BA D203 C52E 43AD 176CE 003AD 8706         MVC   GR2(4),IBLANK
0175C0 D202 C52E C625 176CE 177C5 8707         MVC   GR2+1(3),ICREDIT+1
0175C6 4110 C4FE                 176FE 8708         LA    1,GRM2
0175CA 45E0 D158                 02258 8709         BAL   14,SOSPRINT
0175CE 47F0 C3D6                 17576 8710         B     OUT              LET'S GO SEE WHAT THE STUDENT IS DOING

                                     8712 *         ********************************************************************
                                     8713 *         **********                                              **********
                                     8714 *         **********          STUDENT ANSWER ROUTINE              **********
                                     8715 *         **********                                              **********
                                     8716 *         ********************************************************************

0175D2 1875                      C0000 8718 GETANS  LR    7,5              REG 5 HAS CONTROL COUNT
0175D4 897C 0002                 CCCC2 8719         SLL   7,2              MULTIPLY BY 4
0175D8 5877 0EF8                 C2EF8 8720         L     7,ICORE(7)       LOAD (SOS) AFTER SVC
0175DC 897C 0002                 CCCC2 8721         SLL   7,2              MULTIPLY ADDRESS BY 4
0175E0 4177 0EF8                 C2EF8 8722         LA    7,ICORE(7)       LOAD 360 ADDRESS OF ANSWER
0175E4 1867                      C0000 8723         LR    6,7              SAVE THIS ADDRESS.  IT WILL BE USED TO
                                     8724 *                                SUBSTITUTE IN THE CORRECT ANSWER IN
                                     8725 *                                THE INTERMEDIATE ANSWER GRADING ROUT.
0175E6 5877 0000                 CCCCC 8726         L     7,0(7)           OBTAIN THE STUDENT'S ANSWER
0175EA 507C C614                 177B4 8727         ST    7,STUANS         PUT IN STUDENT ANSWER AREA
0175EE 07FE                      C0000 8728         BR    14               GO BACK TO GRADING ROUTINE

                                     8730 *         ********************************************************************
                                     8731 *         **********                                              **********
                                     8732 *         **********          MAIN GRADER MESSAGES                **********
                                     8733 *         **********                                              **********
                                     8734 *         ********************************************************************

                                     8736 GERR3     GPRT  42,27,01,'  **SVC NOT FOLLOWED BY PROPER PARAMETER**'
0175F0 4110 C950                 17AF0 8737+GERR3   LA    1,=CL42'  **SVC NOT FOLLOWED BY PROPER PARAMETER**'
0175F4 D201 1000 C97A 00000 17B1A 8738+        MVC   0(2,1),=X'2701'
0175FA 58F0 C738                 17BDE 8739+        L     15,=A(SOSPRINT)
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
017FFE 05EF                      8740+          BALR  14,15
018000 47F0 C3D6            17D76 8741           B     OUT
                                   8742 GERR4     GPRT  45,28,01,'  **INCORRECT NO. OF INT. ANSWERS SPECIFIED**'
018004 4110 CC64            17FC4 8743+GERR4     LA    1,=CL45'  **INCORRECT NO. OF INT. ANSWERS SPECIFIED**'
018008 D201 1000 C97C CCCCC 17810 8744+          MVC   0(2,1),=X'2A01'
01800F 58EC C738            17BD8 8745+          L     15,=A(SOSPRINT)
018012 05EF                       8746+          BALR  14,15
018014 47F0 C3D6            17D76 8747           B     OUT
                                   8748 GRM1      GMESS L132'          $$$$END OF GRADER RUN: TOTAL MACHINE GRADE=X
                                                       $$$$',(GR1,52,4),C=2
01801A                             8749+          ORG   *+2
01801A                             8750+GRM1M     DS    0CL132
018018                             8751+          ORG   *-2
018018 83                          8752+GRM1      DC    AL1(131)
018019 C2                          8753+          DC    XL1'2'
                                   8754+          DC    CL132'          $$$$END OF GRADER RUN: TOTAL MACHINE GRADEX
01801A 404C404040404C4C                                +                =     $$$$'
018040                             8755+          ORG   GRM1M+52-1
018040                             8756+GR1       DS    0CL4
01769E                             8757+          ORG
                                   8758 GRM2      GMESS L132'     ###ACCUMULATED INTERMEDIATE ANSWER CREDIT=      X
                                                       ###',(GR2,47,4),C=2
0176AC                             8759+          ORG   *+2
0176AC                             8760+GRM2M     DS    0CL132
01769E                             8761+          ORG   *-2
01769E 83                          8762+GRM2      DC    AL1(131)
01769F C2                          8763+          DC    XL1'2'
                                   8764+          DC    CL132'     ###ACCUMULATED INTERMEDIATE ANSWER CREDIT=     X
0176AC 404040407878787C1                               +   ###'
0176CE                             8765+          ORG   GRM2M+47-1
0176CE                             8766+GR2       DS    0CL4
017724                             8767+          ORG
                                   8768 GRM3      GMESS L132'          $$$$ADDITIONAL CREDIT: TOTAL MACHINE GRADE X
                                                       WITH BONUS POINTS=      $$$$',(GR3,70,4),C=2
017726                             8769+          ORG   *+2
017726                             8770+GRM3M     DS    0CL132
017724                             8771+          ORG   *-2
017724 83                          8772+GRM3      DC    AL1(131)
017725 C2                          8773+          DC    XL1'2'
                                   8774+          DC    CL132'          $$$$ADDITIONAL CREDIT: TOTAL MACHINE GRADEX
017726 404040404040C4C4C                               +   WITH BONUS POINTS=      $$$$'
017748                             8775+          ORG   GRM3M+70-1
017748                             8776+GR3       DS    0CL4
0177AA                             8777+          ORG


0177AA C000
0177AC C0000000                    8779 SUM       DC    F'0'
0177B0 C0C0C0CC                    8780 SUM1      DC    F'0'
0177B4                             8781 STUANS    DS    F
0177B8                             8782 WORKWORD  DS    D
0177C0 C0C00000                    8783 CREDIT    DC    F'0'
0177C4 C0C0C0CC                    8784 ICREDIT   DC    F'0'
0177C8 C0C0C000                    8785 XCREDIT   DC    F'0'
0177CC 83                          8786           DC    AL1(131)
0177CD C2                          8787           DC    XL1'2'
0177CE                             8788 GEND      DS    CL132
017852 5C5CC3C1D3E34CC5            8789 GHALT1    DC    C'**HALT INSTRUCTION:'
01786F C9D5E2E3D9E4C3E3            8790 GHALT2    DC    C'INSTRUCTIONS EXECUTED**'
01787C D200 C642 1000 17FE2 C0CCC  8791 GMVC      MVC   GEND+20(0),0(1)
```

```
   LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
017888                            8792          LTORG
017888 40405C5CE2C5C3D6           8793                =CL40'  **SECOND INTERMEDIATE ANSWER CORRECT**'
017880 40405C5CC6D6E4D5           8794                =CL40'  **FOURTH INTERMEDIATE ANSWER CORRECT**'
017898 C00C2258                   8795                =A(SOSPRINT)
017890 40405C5CC6C9D5C1           8796                =CL28'  **FINAL ANSWER INCORRECT**'
0178E8 C0000005                   8797                =F'5'
0178EC C0000000                   8798                =F'0'
017900 000177E4                   8799                =A(GEND+22)
                                  8800                =CL92'  **YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THAN TX
                                                      HE MEDIAN FOR PROJECTS IN THIS CATEGORY.**'
017904 40405C5CE8D6E4D5
017960 00016E84                   8801                =A(SAVE14)
017964 40404040                   8802                =CL4'    '
017968 C0017CC0                   8803                =A(SAVE214)
01796C 0006                       8804                =H'6'
01796E 0001                       8805                =H'1'
017970 2401                       8806                =X'2401'
017972 4E01                       8807                =X'4E01'
017974 5202                       8808                =X'5202'
017976 2501                       8809                =X'2501'
                                  8810                =CL92'  **SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA SX
                                                      UBSTITUTED. ROUTINE CONTINUED.**'
017978 40405C5CE2C5C3D6
017CCA 4E01                       8811                =X'4E01'
                                  8812                =CL86'  **SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA NX
                                                      OT SUBSTITUTED. ROUTINE CONTINUED.**'
017CCC 40405C5CE2C5C3D6
017A22 5302                       8813                =X'5302'
                                  8814                =CL32'  **FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATA SX
                                                      UBSTITUTED. ROUTINE CONTINUED.**'
017A24 40405C5CC6D6E4D5
                                  8815                =CL86'  **FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATA NX
                                                      OT SUBSTITUTED. ROUTINE CONTINUED.**'
017A76 40405C5CC6D6E4D5
017ACC 40405C5CC6C9D5C1           8816                =CL26'  **FINAL ANSWER CORRECT**'
017AE6 1702                       8817                =X'1702'
017AE8 1902                       8818                =X'1902'
017AEA 7C02                       8819                =X'7C02'
017AEC 5902                       8820                =X'5902'
017AEE C102                       8821                =X'0102'
017AF0 40405C5CE2F5C340           8822                =CL42'  **SVC NOT FOLLOWED BY PROPER PARAMETER**'
017B1A 2701                       8823                =X'2701'
017B1C 2A01                       8824                =X'2A01'
017B1E 40405C5CC6C9D9E2           8825                =CL39'  **FIRST INTERMEDIATE ANSWER CORRECT**'
                                  8826                =CL81'  **FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUX
                                                      BSTITUTED. ROUTINE CONTINUED.**'
017B45 40405C5CC6C9D9E2
                                  8827                =CL85'  **FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA NCX
                                                      T SUBSTITUTED. ROUTINE CONTINUED.**'
017B96 40405C5CC6C9D9E2
017BEB 40405C5CE3C8C9D9           8828                =CL39'  **THIRD INTERMEDIATE ANSWER CORRECT**'
                                  8829                =CL81'  **THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA SUX
                                                      BSTITUTED. ROUTINE CONTINUED.**'
017C12 40405C5CE3C8C9D9
                                  8830                =CL85'  **THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA NCX
                                                      T SUBSTITUTED. ROUTINE CONTINUED.**'
017C63 40405C5CE3C8C9D9
017C8C 40405C5CC6C9C6E3           8831                =CL39'  **FIFTH INTERMEDIATE ANSWER CORRECT**'
                                  8832                =CL81'  **FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATA SUX
                                                      BSTITUTED. ROUTINE CONTINUED.**'
017CDE 40405C5CC6C9C6E3
                                  8833                =CL85'  **FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATA NCX
                                                      T SUBSTITUTED. ROUTINE CONTINUED.**'
017D30 40405C5CC6C9C6E3
                                  8834                =CL127'  **YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS THANX
                                                       THE MEDIAN FOR PROJECTS IN THIS CATEGORY. ADDITIONAL CRX
                                                      EDIT IS GIVEN BELOW.**'
017D85 40405C5CE8D6E4D5
017E04 40405C5CC9D5C3D6           8835                =CL45'  **INCORRECT NO. OF INT. ANSWERS SPECIFIED**'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                 8833 *        ****************************************************************
                                 8839 *        ****************************************************************
                                 8840 *        **********                                            **********
                                 8841 *        **********                                            **********
                                 8842 *        **********      GENERAL COMMUNICATIONS SECTION        **********
                                 8843 *        **********                                            **********
                                 8844 *        **********                                            **********
                                 8845 *        ****************************************************************
                                 8846 *        ****************************************************************

000000                           8848 GINITDS   DSECT
000000                           8849 ANS1      DS    F
000004                           8850 ANS2      DS    F
000008                           8851 ANS3      DS    F
00000C                           8852 ANS4      DS    F
000010                           8853 ANS5      DS    F
000014                           8854 ANSF      DS    F
000018                           8855 NOINTANS  DS    F
00001C                           8856 NCX       DS    F
000020                           8857 CONTROL   DS    H
                                 8858 IMESS1    GMESS L132'     '
000024                           8859+          ORG   *+2
000024                           8860+IMESS1M   DS    0CL132
000022                           8861+          ORG   *-2
000022 83                        8862+IMESS1    DC    AL1(131)
000023 01                        8863+          DC    XL1'01'
000024 404040404040404040        8864+          DC    CL132'     '
000088                           8865+          ORG
                                 8866 IMESS2    GMESS L132'     '
0000AA                           8867+          ORG   *+2
0000AA                           8868+IMESS2M   DS    0CL132
0000A8                           8869+          ORG   *-2
0000A8 83                        8870+IMESS2    DC    AL1(131)
0000A9 01                        8871+          DC    XL1'01'
0000AA 404040404040404040        8872+          DC    CL132'     '
00012E                           8873+          ORG
                                 8874 IMESS3    GMESS L132'     '
000130                           8875+          ORG   *+2
000130                           8876+IMESS3M   DS    0CL132
00012E                           8877+          ORG   *-2
00012E 83                        8878+IMESS3    DC    AL1(131)
00012F 01                        8879+          DC    XL1'01'
000130 404040404040404040        8880+          DC    CL132'     '
0001B4                           8881+          ORG
                                 8882 IMESS4    GMESS L132'     '
0001B6                           8883+          ORG   *+2
0001B6                           8884+IMESS4M   DS    0CL132
0001B4                           8885+          ORG   *-2
0001B4 83                        8886+IMESS4    DC    AL1(131)
0001B5 01                        8887+          DC    XL1'01'
0001B6 404040404040404040        8888+          DC    CL132'     '
00023A                           8889+          ORG
                                 8890 IMESS5    GMESS L132'     '
00023C                           8891+          ORG   *+2
00023C                           8892+IMESS5M   DS    0CL132
00023A                           8893+          ORG   *-2
00023A 83                        8894+IMESS5    DC    AL1(131)
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000238 C1                        8895+          DC     XL1'01'
000239 404040404040404C          8896+          DC     CL132'     '
0002CC                           8897+          ORG
                                 8898 IMESSF    GMESS  L132'     '
0002C2                           8899+          ORG    *+2
0002C2                           8900+IMESSFM   DS     0CL132
0002C0                           8901+          ORG    *-2
0002CC 83                        8902+IMESSF    DC     AL1(131)
0002C1 C1                        8903+          DC     XL1'01'
0002C2 404040404040404C          8904+          DC     CL132'     '
000346                           8905+          ORG
```

APPENDIX C

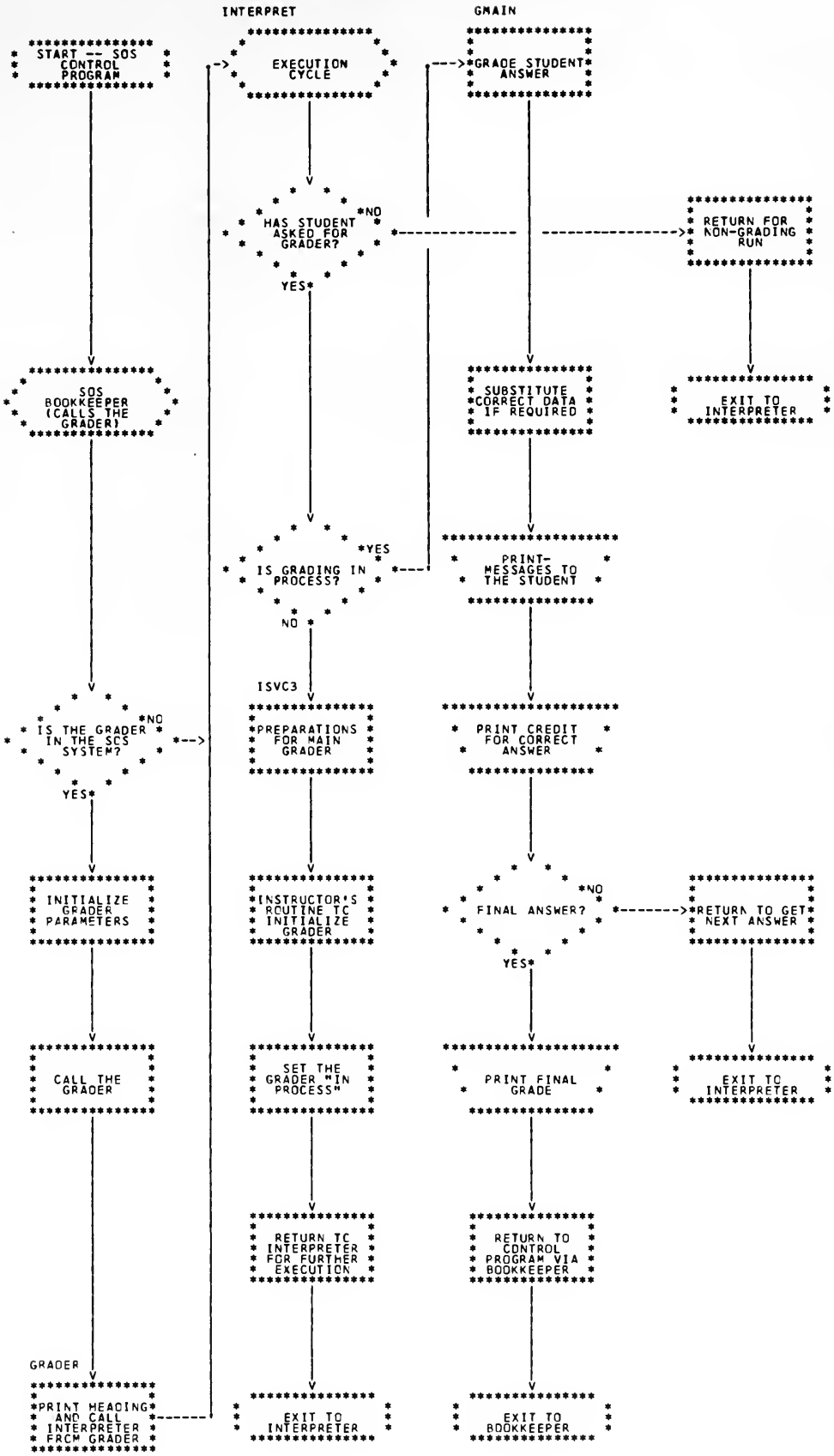
Flowchart Documentation of the Grader Program

The flowcharts which document the various sections of the Grader Program are the following:

Flowchart Number	Title
1.	The Basic Grading Routine
2.	Grader/SOS Interface and Grader Initialization
3.	Intermediate Answer Grading Routine
4.	Final answer Grading Routine
5.	Routine for Accumulating Intermediate Answer Credit
6.	Routine to Obtain Student's Answer
7.	Translation and Printout of Machine Grade Routine
8.	Return Routines
9.	Total Machine Grade Routine (First Section)
10.	Total Machine Grade Routine (Second Section)
11.	Instructions Executed Routine
12.	Structure of General Communications Section

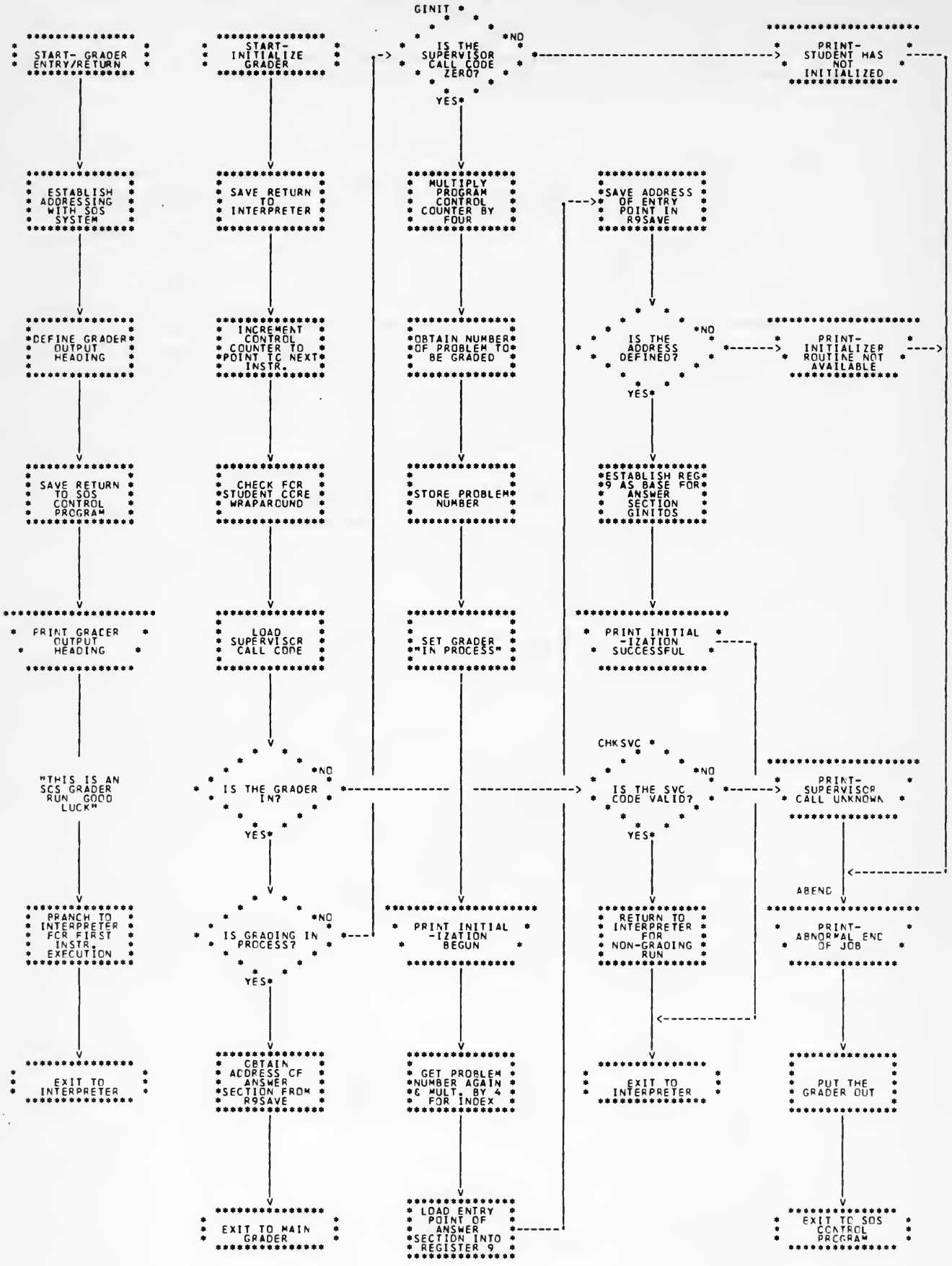
*** APPENDIX C -- CHART NUMBER 1 ***

FLOW CHART OF THE BASIC GRADING ROUTINE



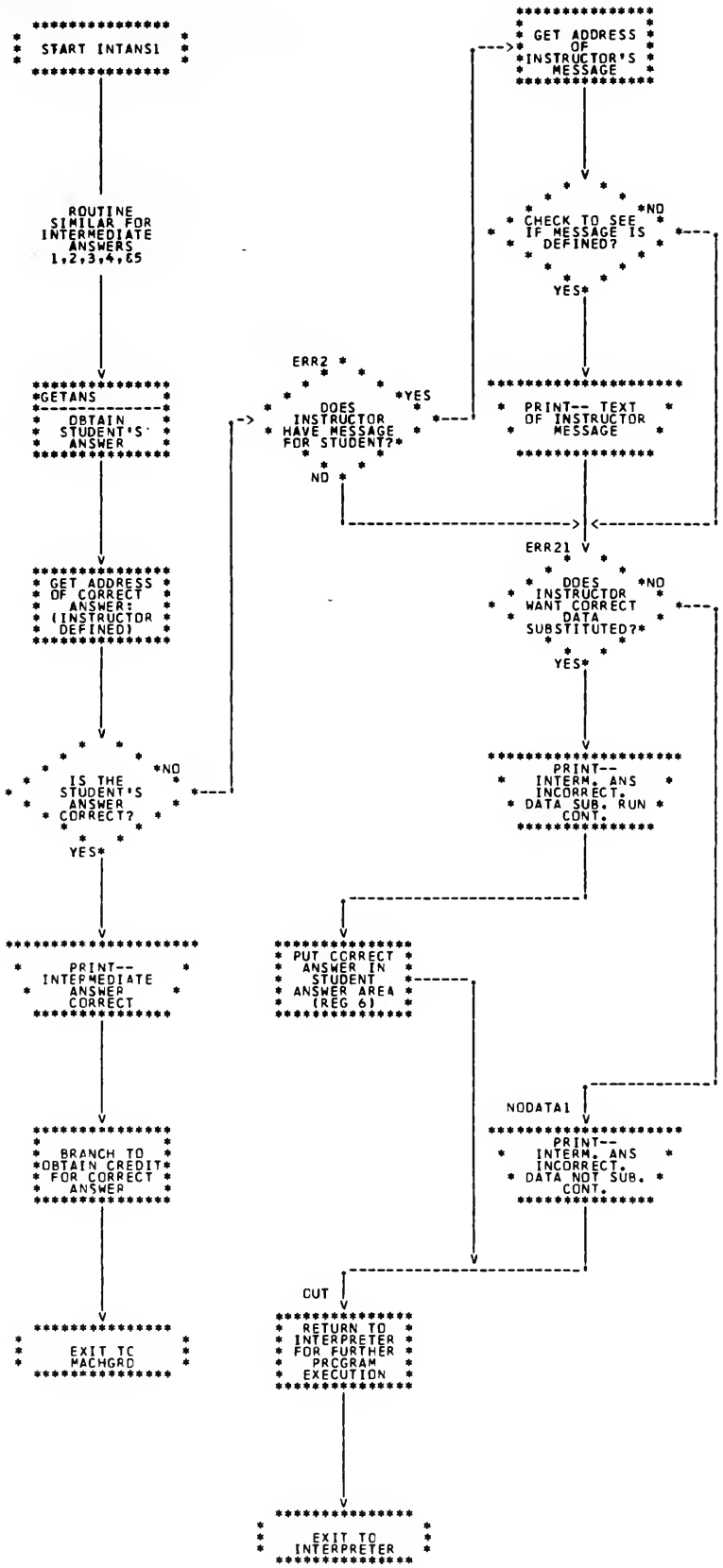
*** APPENDIX C -- CHART NUMBER 2 ***

GRADER/SOS INTERFACE AND GRADER INITIALIZATION



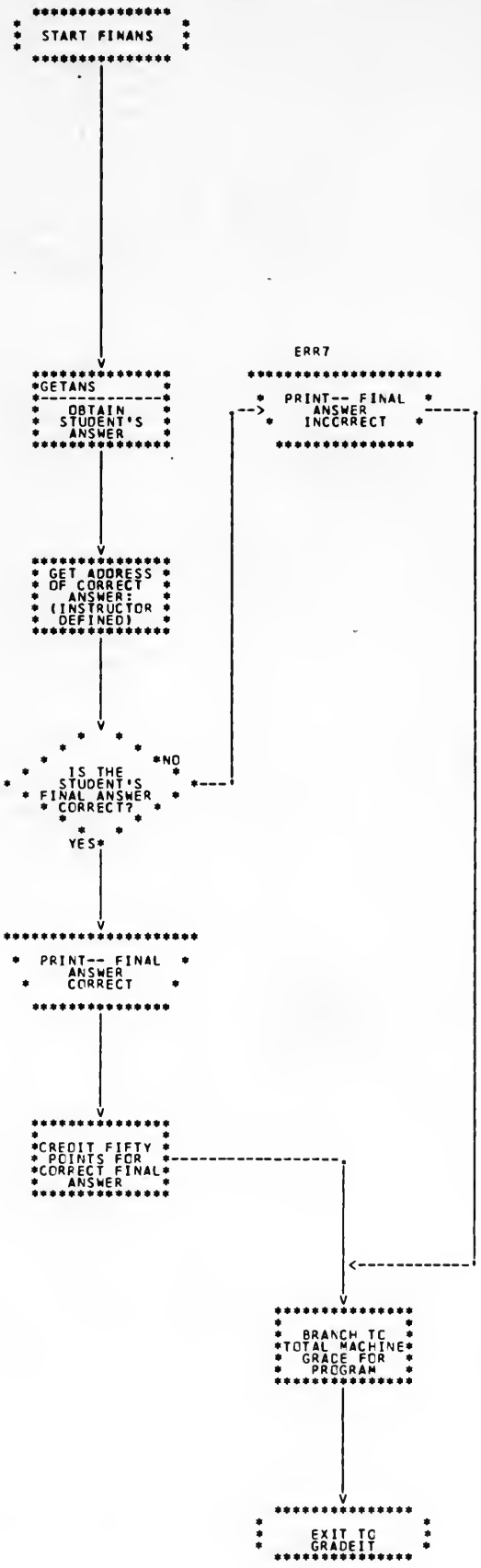
*** APPENDIX C -- CHART NUMBER 3 ***

INTERMEDIATE ANSWER GRADING ROUTINE



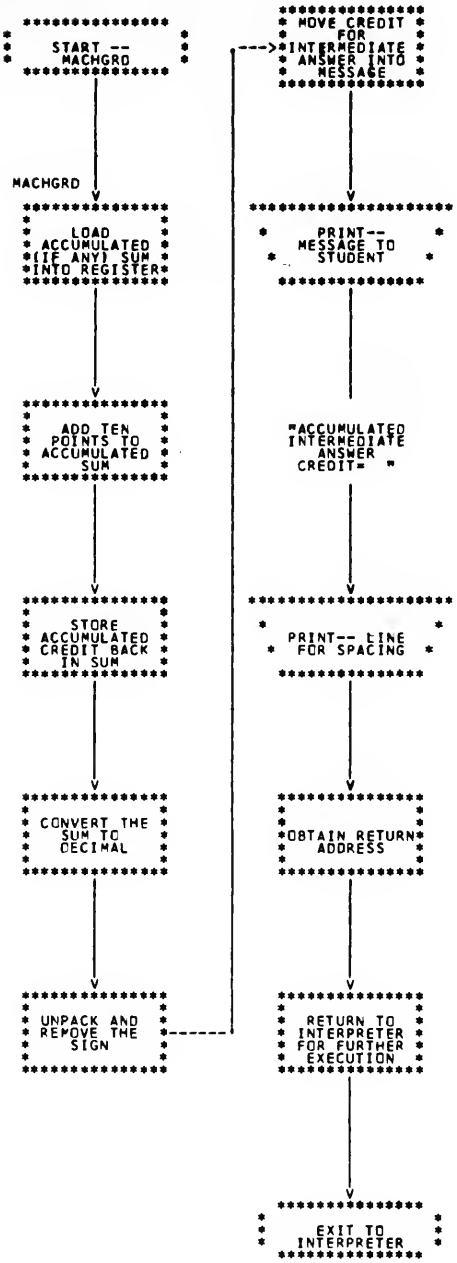
*** APPENDIX C -- CHART NUMBER 4 ***

FINAL ANSWER GRADING ROUTINE



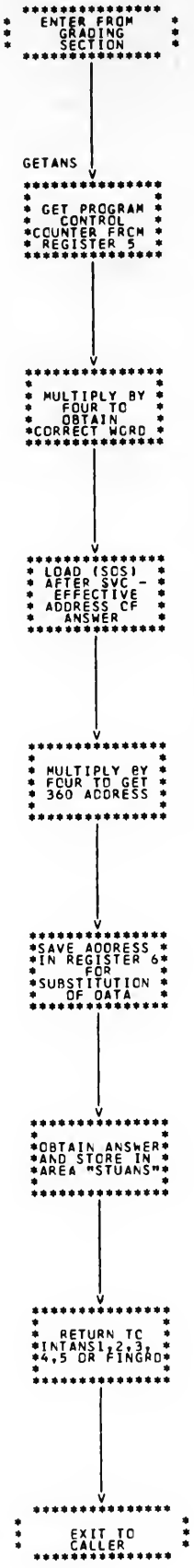
*** APPENDIX C -- CHART NUMBER 5 ***

ROUTINE FOR ACCUMULATING INTERMEDIATE ANSWER CREDIT



*** APPENDIX C -- CHART NUMBER 6 ***

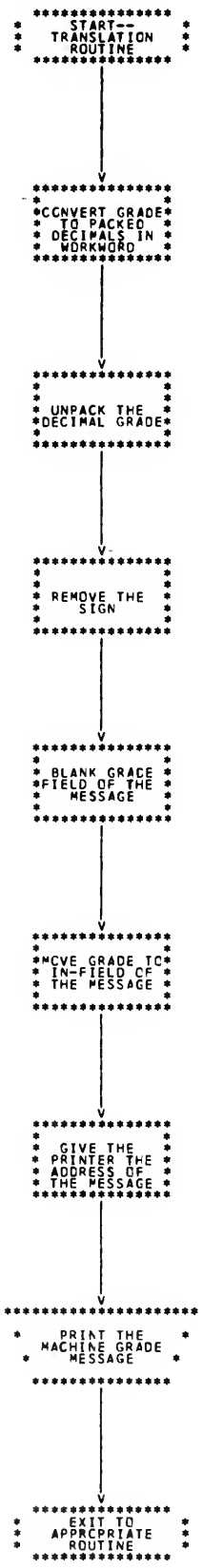
ROUTINE TO OBTAIN STUDENT'S ANSWER



*** APPENDIX C -- CHART NUMBER 7 ***

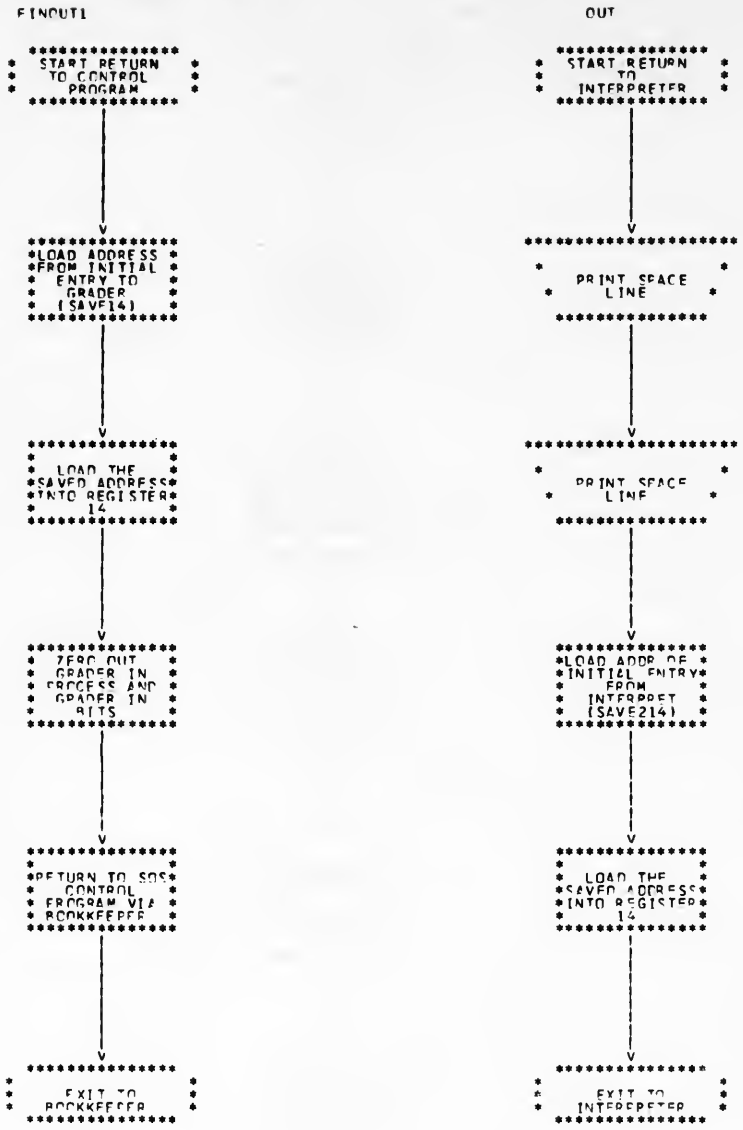
TRANSLATION AND PRINTOUT OF MACHINE GRADE ROUTINE

TRANS1,2,3



*** APPENDIX C -- CHART NUMBER 8 ***

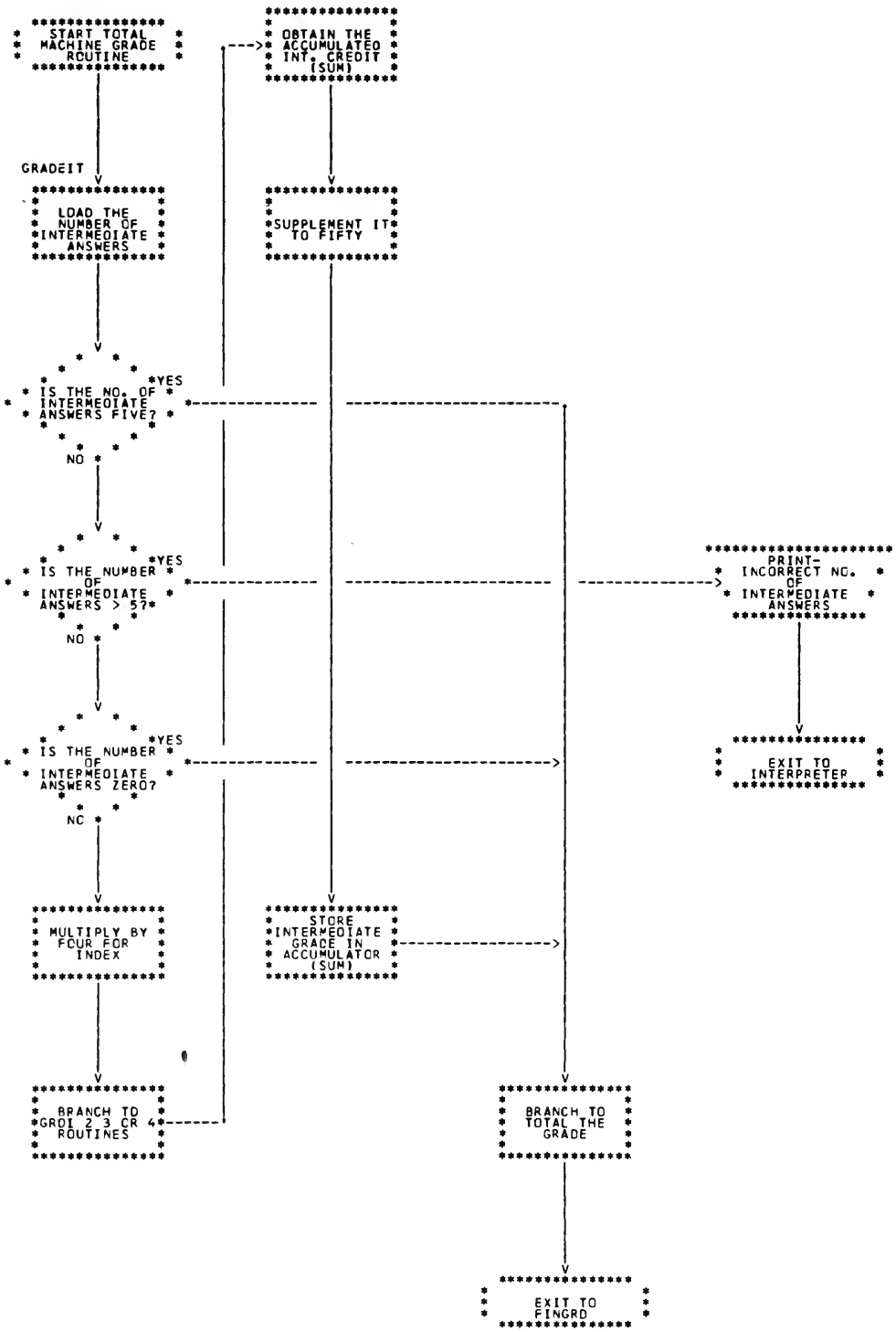
RETURN ROUTINES



24C814

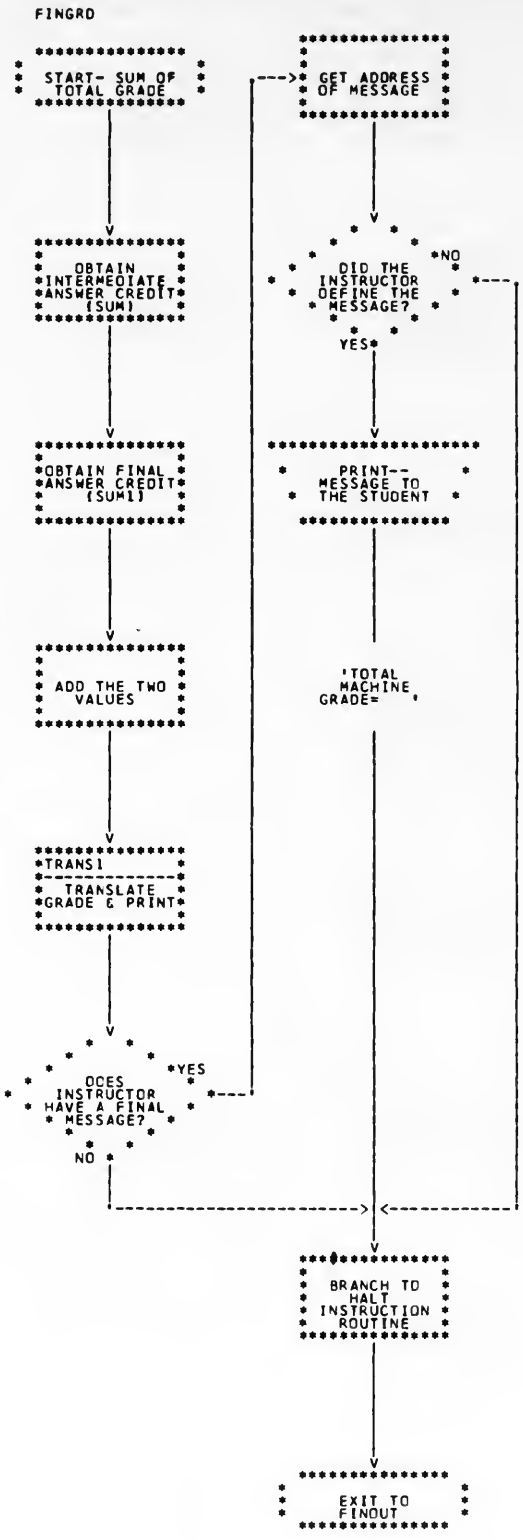
*** APPENDIX C -- CHART NUMBER 9 ***

TOTAL MACHINE GRADE ROUTINE (FIRST SECTION)



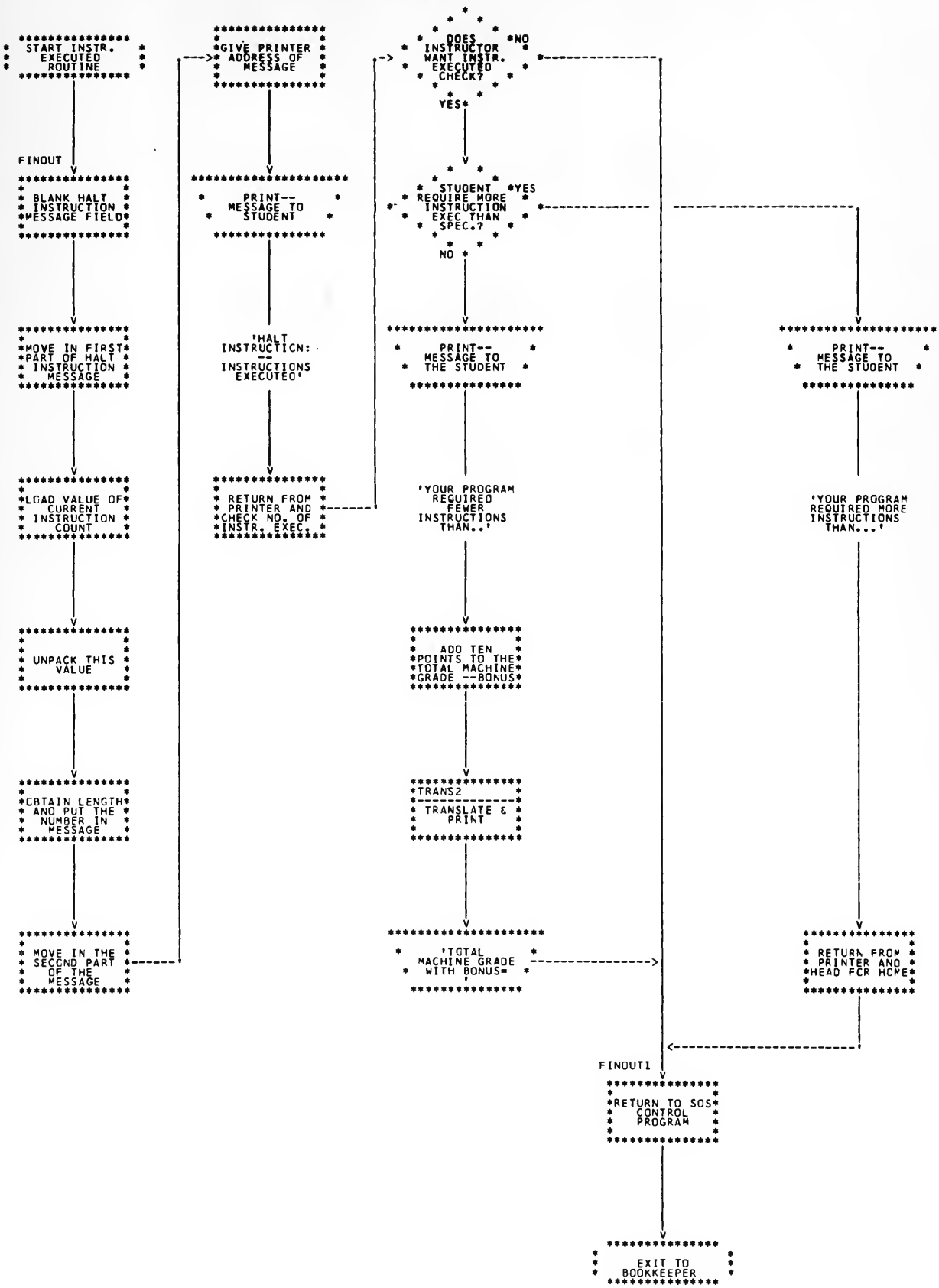
*** APPENDIX C -- CHART NUMBER 10 ***

TOTAL MACHINE GRADE ROUTINE (SECOND SECTION)



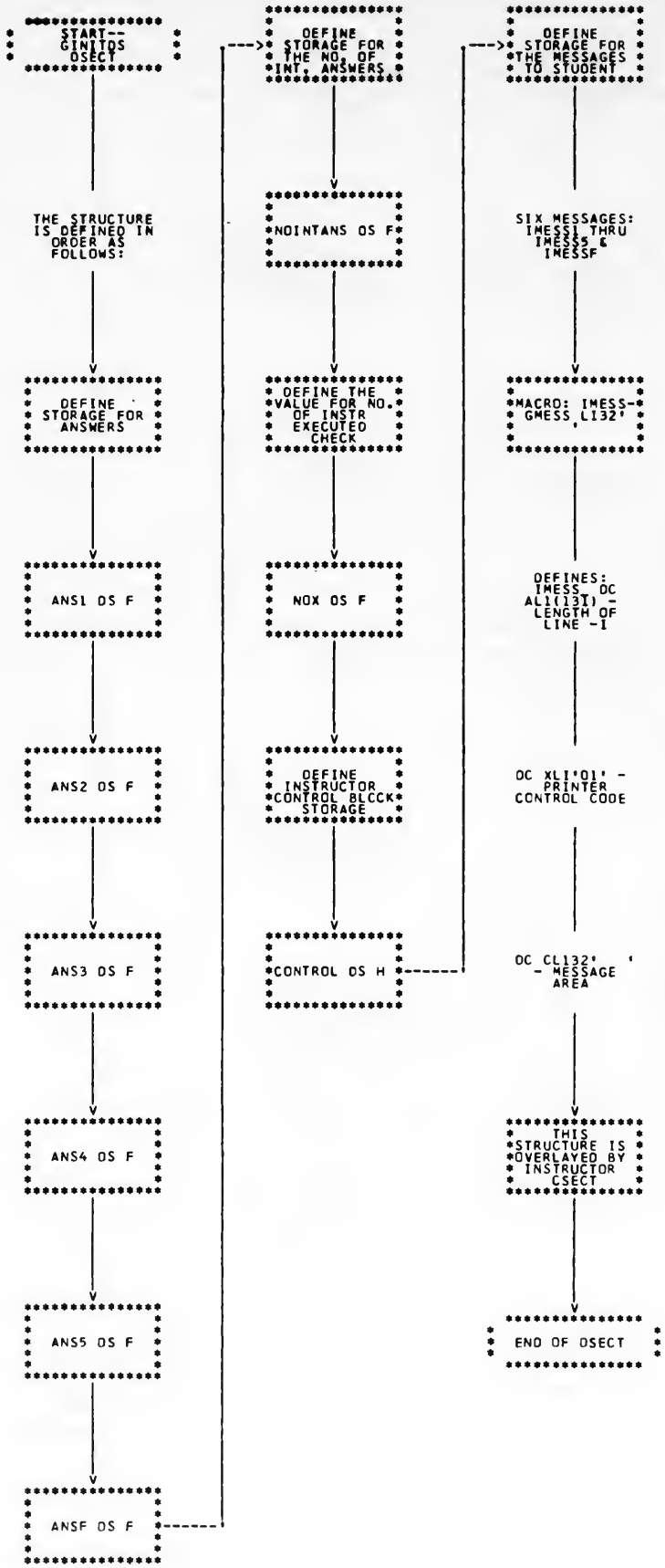
*** APPENDIX C -- CHART NUMBER 11 ***

INSTRUCTIONS EXECUTED ROUTINE



*** APPENDIX C -- CHART NUMBER 12 ***

STRUCTURE OF GENERAL COMMUNICATIONS SECTION



APPENDIX D

Examples of Graded Programs

An example of a program with the grader off is first shown. Examples of graded programs are then given.

```
NAME: BACONR20  JOBNAME: EXAMPL20  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
BOOKKEEPING PHASE MAP OF OPTIONS

       CONFIGURATION--1,
       CORE ZEROED--YES,
       DUMP REQUESTED--YES,
       GRADER--NO,
      1MAX NUM EXECUTABLE INSTRUCTIONS-SOSASUM(3),
      1MAX NUM PRINTED LINES--SOSASUM(2),
      1MAX NUM TRACED INSTRUCTIONS--SOSASUM(1),
       OUTPUT HEADER--TRACE,
       STATUS--STUDENT,
       TRACE INITIALLY--ON,
       WARNINGS PRINTED--YES,

    1SOSASUM=(500,1000,2000)
    2SOSMAX=(200,1000,10000)

**************************************
*                                    *
*IF A MAN WILL BEGIN WITH CERTAINTIES*
*                                    *
*HE SHALL END IN DOUBTS; BUT IF HE   *
*                                    *
*WILL BE CONTENT TO BEGIN WITH DOUBTS*
*                                    *
*HE SHALL END IN CERTAINTIES.        *
*                                    *
*                       --  BACON    *
**************************************

NO EDITING ERRORS OCCURRED DURING EDIT PHASE
PROGRAM EXAMPL20 WAS NOT MODIFIED
NEXT RUN IS NO.  2
EDIT PHASE COMPLETED---SUCCESSFULLY
```

```
LOCATION  OBJECT    STATEMENT     CARD IMAGE
COUNTER    CODE      NUMBER
                        1       ********************************
                        2       ********************************
                        3       **                            **
                        4       **  DEMONSTRATION OF GRADER   **
                        5       **                            **
                        6       ********************************
                        7       ********************************
                        8       *
                        9       *
                       10       * NOTE 1: EACH ANSWER DEPENDS
                       11       * ON PREVIOUS RESULTS.
                       12       *
                       13       * NOTE 2: THE PROBLEM USES
                       14       * "IMMEDIATE" TYPE DATA.
                       15       *
                       16       * NOTE 3: THE PROBLEM IS AN
                       17       * EXAMPLE OF GRADER INTER-
                       18       * ACTION ONLY AND IS NOT CON-
                       19       * SIDERED TO BE SUITABLE FOR
                       20       * A STUDENT TYPE PROBLEM.
                       21       *
                       22       ********************************
                       23       *                              *
                       24       *       FIRST ANSWER           *
                       25       *                              *
                       26       ********************************
                       27       *
                       28       * THE FIRST TWO INSTRUCTIONS
                       29       * INITIALIZE THE GRADER FOR
                       30       * PROBLEM NO. 1. "TOFF" TURNS
                       31       * THE TRACE OFF. THE NUMBERS
                       32       * 125 AND 24 ARE THEN ADDED
                       33       * AND 50 IS SUBTRACTED IN THE
                       34       * EXECUTION OF THE SINGLE
                       35       * INSTRUCTION "AXAI".
                       36       *
                       37       *
                       38       *
 010    C0300000       39       BEGIN    SVC   3,0
 011    00000000       40                DC    0
                       41                TOFF  0
 015    A133007D       42                SXAI  3,125(3)
 016    A144C018       43                SXAI  4,24(4)
 017    A234FFCE       44                AXAI  3,-50(4)
 018    60300064       45                ST    3,NUMBRS
                       46       *
                       47       *
                       48       *
                       49       * THE RESULT IS THEN PRINTED
                       50       * AND THE GRADER IS CALLED TO
                       51       * CHECK THE FIRST INTERMEDIATE
                       52       * ANSWER WHICH IS LOCATED AT
                       53       * THE EFFECTIVE ADDRESS OF
                       54       * "NUMBRS".
                       55       *
                       56                PUTC  NUMBRS,NO=10
                       57                RET
 020    C0300001       58                SVC   3,1
 021    00000064       59                H     NUMBRS
                       60       *
```

```
LOCATION  OBJECT    STATEMENT     CARD IMAGE
COUNTER    CODE      NUMBER
                       61        *
                       62        *********************************
                       63        *                               *
                       64        *        SECOND ANSWER          *
                       65        *                               *
                       66        *********************************
                       67        *
                       68        * THE NEXT SECTION ADDS "NUMBRS"
                       69        * NUMBRS+1 TIMES. THE RESULT
                       70        * IS PRINTED AND THE GRADER
                       71        * IS CALLED TO INSPECT THE 2ND
                       72        * INTERMEDIATE ANSWER.
                       73        *
                       74        *
                       75        *
                       76        *
 C22    5C500064       77                 L      5,NUMBRS
 C26    1C500064       78        LOOP1    A      5,NUMBRS
 C24    AC400023       79                 BCT    4,LOOP1
 C25    6C500064       80                 ST     5,NUMBRS
                       81                 PUTD   NUMBRS,NO=10
                       82                 RET
 C2D    CC30C002       83                 SVC    3,2
 C2E    C0000064       84                 H      NUMBRS
                       85        *
                       86        *
                       87        *********************************
                       88        *                               *
                       89        *       THIRD ANSWER            *
                       90        *                               *
                       91        *********************************
                       92        *
                       93        * THE RESULT OF THE PREVIOUS
                       94        * SECTION IS THEN MULTIPLIED
                       95        * BY 16 AND THEN DIVIDED BY
                       96        * FOUR. THE GRADER THEN CHECKS
                       97        * THE THIRD ANSWER.
                       98        *
                       99        *
 C2F    50600064      100                 L      6,NUMBRS
 C30    806CC004      101                 SLL    6,4
 C31    716CC002      102                 SRA    6,2
 C32    60600064      103                 ST     6,NUMBRS
                      104                 PUTD   NUMBRS,NO=10
                      105                 RET
 C3A    C0300003      106                 SVC    3,3
 C3B    C0000064      107                 H      NUMBRS
                      108        *
                      109        *********************************
                      110        *                               *
                      111        *        FOURTH ANSWER          *
                      112        *                               *
                      113        *********************************
                      114        *
                      115        * A DIVISION AND MULTIPLICATION
                      116        * THEN OCCUR AND THE FOURTH
                      117        * ANSWER IS GRADED.
                      118        *
                      119        *
 03C    20600006      120                 S      6,6
```

```
LOCATION  OBJECT    STATEMENT     CARD IMAGE
COUNTER     CODE     NUMBER
  C3D   5C70C064       121              L     7,NUMBRS
  C3E   40600064       122              D     6,NUMBRS
  C3F   3060006E       123              M     6,=16
  C40   60700064       124              ST    7,NUMBRS
                       125              PUTD  NUMBRS,NO=10
                       126              RET
  048   C0300004       127              SVC   3,4
  C4S   C0C00064       128              H     NUMBRS
                       129     ********************************
                       130     *                              *
                       131     *         FIFTH  ANSWER        *
                       132     *                              *
                       133     ********************************
                       134     *
                       135     * THE RESULT OF THE PREVIOUS
                       136     * SECTION IS THEN MADE
                       137     * EQUAL TO THE MAXIMUM VALUE
                       138     * FOR A DECIMAL CONSTANT AND THE
                       139     * GRADER IS CALLED FOR THE CHECK
                       140     * OF THE LAST INTERMEDIATE
                       141     * ANSWER.
                       142     *
                       143     *
  04A   50300064       144              L     3,NUMBRS
  04B   1030006F       145              A     3,=2147483631
  04C   60300064       146              ST    3,NUMBRS
                       147              PUTD  NUMBRS,NO=10
                       148              RET
  054   C0300005       149              SVC   3,5
  055   00000064       150              H     NUMBRS
                       151     *
                       152     *
                       153     *
                       154     ********************************
                       155     *                              *
                       156     *           FINAL ANSWER       *
                       157     *                              *
                       158     ********************************
                       159     *
                       160     *
  056   20500005       161              S     5,5
  057   50900064       162              L     9,NUMBRS
  058   A159FFFF       163              SXAI  5,-1(9)
  059   60500064       164              ST    5,NUMBRS
                       165              PUTD  NUMBRS,NO=11
                       166              RET
  C61   C0300006       167              SVC   3,6
  062   00000064       168              H     NUMBRS
  C63   C0C0C000       169     EOF      H     0
  064                  170     NUMBRS   DS    10
  C1C                  171              END   BEGIN
***LITERAL ALLOCATION FOLLOWS***

  06E   C0CC0010
  C6F   7FFFFFEF
```

```
NAME: BACONR20  JCBNAME: EXAMPL20  NAVAL PCSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969

SYMBCL TABLE
SYMBCL   DEFINITION
BEGIN    00000010
EOF      0C000063
LCCP1    C0000023
NUMBRS   00000064
***NO  CCMPLETELY UNASSEMBLED STATEMENTS***
***ASSEMBLY PHASE COMPLETED SUCCESSFULLY***
**LCAD PHASE CCMPLETED SUCCESSFULLY**
```

```
NAME: BACONR20  JOBNAME: EXAMPL20  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360     29 MAY, 1969
 CONTROL    CURRENT    REG        OLD REG          NEW REG        XREG     INDEX REG        IDADD   EFF      OLD WORD    NEW WORD
 COUNTER     INST      NOS        CONTENTS         CONTENTS        NO      CONTENTS         LEVEL   ADD      CONTENTS    CONTENTS
   010      C03C0000                                                                                000
                                                                                                    014
   012      90C00014

        99
      2475
      9900
        16
2147483647
-2147483648
**HALT INSTRUCTION: 104 INSTRUCTIONS EXECUTED**
**NORMAL EXIT--END OF JOB**
```

```
NAME: BACONR21  JOBNAME: EXAMPL21  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
BOOKKEEPING PHASE MAP OF OPTIONS

      CONFIGURATION--1,
      CORE ZEROED--YES,
      DUMP REQUESTED--YES,
      GRADER--YES,
     1MAX NUM EXECUTABLE INSTRUCTIONS-SOSASUM(3),
     1MAX NUM PRINTED LINES--SOSASUM(2),
     1MAX NUM TRACED INSTRUCTIONS--SOSASUM(1),
      OUTPUT HEADER--TRACE,
      STATUS--STUDENT,
      TRACE INITIALLY--ON,
      WARNINGS PRINTED--YES,

   1SOSASUM=(500,1000,2000)
   2SOSMAX=(200,1000,10000)

**************************************
*                                    *
*IF A MAN WILL BEGIN WITH CERTAINTIES*
*                                    *
*HE SHALL END IN DOUBTS; BUT IF HE   *
*                                    *
*WILL BE CONTENT TO BEGIN WITH DOUBTS*
*                                    *
*HE SHALL END IN CERTAINTIES.        *
*                                    *
*                     --  BACON      *
**************************************

NO EDITING ERRORS OCCURRED DURING EDIT PHASE
PROGRAM EXAMPL21 WAS NOT MODIFIED
NEXT RUN IS NO.  2
EDIT PHASE COMPLETED---SUCCESSFULLY
```

```
LOCATION  OBJECT    STATEMENT      CARD IMAGE
COUNTER   CODE      NUMBER
                        1       ********************************
                        2       ********************************
                        3       **                            **
                        4       **  DEMONSTRATION OF GRADER   **
                        5       **                            **
                        6       ********************************
                        7       ********************************
                        8       *
                        9       *
                       10       * NOTE 1: EACH ANSWER DEPENDS
                       11       * ON PREVIOUS RESULTS.
                       12       *
                       13       * NOTE 2: THE PROBLEM USES
                       14       * "IMMEDIATE" TYPE DATA.
                       15       *
                       16       * NOTE 3: THE PROBLEM IS AN
                       17       * EXAMPLE OF GRADER INTER-
                       18       * ACTION ONLY AND IS NOT CON-
                       19       * SIDERED TO BE SUITABLE FOR
                       20       * A STUDENT TYPE PROBLEM.
                       21       *
                       22       ********************************
                       23       *                              *
                       24       *      FIRST ANSWER            *
                       25       *                              *
                       26       ********************************
                       27       *
                       28       * THE FIRST TWO INSTRUCTIONS
                       29       * INITIALIZE THE GRADER FOR
                       30       * PROBLEM NO. 1. "TOFF" TURNS
                       31       * THE TRACE OFF. THE NUMBERS
                       32       * 125 AND 24 ARE THEN ADDED
                       33       * AND 50 IS SUBTRACTED IN THE
                       34       * EXECUTION OF THE SINGLE
                       35       * INSTRUCTION "AXAI".
                       36       *
85                     37       *
                       38       *
 010   C03C0000        39       BEGIN   SVC   3,0
 011   00000000        40               DC    0
                       41               TOFF  0
 015   A133C07D        42               SXAI  3,125(3)
 016   A1440018        43               SXAI  4,24(4)
 017   A234FFCE        44               AXAI  3,-50(4)
 018   60300064        45               ST    3,NUMBRS
                       46       *
                       47       *
                       48       *
                       49       * THE RESULT IS THEN PRINTED
                       50       * AND THE GRADER IS CALLED TO
                       51       * CHECK THE FIRST INTERMEDIATE
                       52       * ANSWER WHICH IS LOCATED AT
                       53       * THE EFFECTIVE ADDRESS OF
                       54       * "NUMBRS".
                       55       *
                       56               PUTD  NUMBRS,NO=10
                       57               RET
 020   CC300001        58               SVC   3,1
 021   00000064        59               H     NUMBRS
                       60       *
```

```
LOCATION  OBJECT    STATEMENT     CARD IMAGE
COUNTER   CODE      NUMBER
                      61        *
                      62        *********************************
                      63        *                               *
                      64        *        SECOND ANSWER          *
                      65        *                               *
                      66        *********************************
                      67        *
                      68        * THE NEXT SECTION ADDS "NUMBRS"
                      69        * NUMBRS+1 TIMES. THE RESULT
                      70        * IS PRINTED AND THE GRADER
                      71        * IS CALLED TO INSPECT THE 2ND
                      72        * INTERMEDIATE ANSWER.
                      73        *
                      74        *
                      75        *
                      76        *
  022   50500064      77                  L     5,NUMBRS
  023   1A500064      78        LOOP1     A     5,NUMBRS
  024   A0400023      79                  BCT   4,LOOP1
  025   60500064      80                  ST    5,NUMBRS
                      81                  PUTD  NUMBRS,ND=10
                      82                  RET
  02D   0C30C002      83                  SVC   3,2
  02E   00000064      84                  H     NUMBRS
                      85        *
                      86        *
                      87        *********************************
                      88        *                               *
                      89        *       THIRD ANSWER            *
                      90        *                               *
                      91        *********************************
                      92        *
                      93        * THE RESULT OF THE PREVIOUS
                      94        * SECTION IS THEN MULTIPLIED
                      95        * BY 16 AND THEN DIVIDED BY
                      96        * FOUR. THE GRADER THEN CHECKS
                      97        * THE THIRD ANSWER.
                      98        *
                      99        *
  02F   50600064     100                  L     6,NUMBRS
  030   80600004     101                  SLL   6,4
  031   71600002     102                  SRA   6,2
  032   60600064     103                  ST    6,NUMBRS
                     104                  PUTD  NUMBRS,ND=10
                     105                  RET
  03A   00300003     106                  SVC   3,3
  03B   00000064     107                  H     NUMBRS
                     108       *
                     109       *********************************
                     110       *                               *
                     111       *        FOURTH ANSWER          *
                     112       *                               *
                     113       *********************************
                     114       *
                     115       * A DIVISION AND MULTIPLICATION
                     116       * THEN OCCUR AND THE FOURTH
                     117       * ANSWER IS GRADED.
                     118       *
                     119       *
  03C   20600006     120                  S     6,6
```

```
LOCATION  OBJECT    STATEMENT     CARD IMAGE
COUNTER    CODE     NUMBER
  C3D   507C0064      121                L     7,NUMBRS
  C3E   406C0064      122                C     6,NUMBRS
  C3F   3060006E      123                M     6,=16
  C4C   60700064      124                ST    7,NUMBRS
                      125                PUTD  NUMBRS,NO=10
                      126                RET
  048   C0300004      127                SVC   3,4
  C49   C0000064      128                H     NUMBRS
                      129      ********************************
                      130      *                              *
                      131      *         FIFTH  ANSWER        *
                      132      *                              *
                      133      ********************************
                      134      *
                      135      * THE RESULT OF THE PREVIOUS
                      136      * SECTION IS THEN MADE
                      137      * EQUAL TO THE MAXIMUM VALUE
                      138      * FOR A DECIMAL CONSTANT AND THE
                      139      * GRADER IS CALLED FOR THE CHECK
                      140      * OF THE LAST INTERMEDIATE
                      141      * ANSWER.
                      142      *
                      143      *
  04A   50300064      144                L     3,NUMBRS
  C4B   1030006F      145                A     3,=2147483631
  C4C   60300064      146                ST    3,NUMBRS
                      147                PUTD  NUMBRS,NO=10
                      148                RET
  C54   C0300005      149                SVC   3,5
  C55   00000064      150                H     NUMBRS
                      151      *
                      152      *
                      153      *
                      154      ********************************
                      155      *                              *
                      156      *           FINAL ANSWER       *
                      157      *                              *
                      158      ********************************
                      159      *
                      160      *
  C56   20500005      161                S     5,5
  C57   50900064      162                L     9,NUMBRS
  C58   A159FFFF      163                SXAI  5,-1(9)
  C59   60500064      164                ST    5,NUMBRS
                      165                PUTD  NUMBRS,NO=11
                      166                RET
  061   C0300006      167                SVC   3,6
  C62   00000064      168                H     NUMBRS
  C63   CCCC0000      169      EOF       H     0
  064                 170      NUMBRS    DS    10
  C10                 171                END   BEGIN
***LITERAL ALLOCATION FOLLOWS***

  06E   C0000010
  C6F   7FFFFFEF
```

```
NAME: BACONR21  JOBNAME: EXAMPL21  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360     29 MAY, 1969

SYMBOL TABLE
SYMBOL   DEFINITION
BEGIN    00000010
ECF      00000063
LOOP1    00000023
NUMBRS   00000064
***NO   COMPLETELY UNASSEMBLED STATEMENTS***
***ASSEMBLY PHASE COMPLETED SUCCESSFULLY***
**LOAD PHASE COMPLETED SUCCESSFULLY**
```

```
NAME: BACONR21  JOBNAME: EXAMPL21  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
THIS IS AN SOS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


        99
**FIRST INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 010 ###


      2475
**SECOND INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 020 ###


      9900
**THIRD INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 030 ###


        16
**FOURTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 040 ###


2147483647
**FIFTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 050 ###

89

-2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 100 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 103 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY. ADDITIONAL CREDIT IS GIVEN BELOW.**
        $$$$ADDITIONAL CREDIT: TOTAL MACHINE GRADE WITH BONUS POINTS= 110 $$$$
```

```
NAME: BACONR22  JOBNAME: EXAMPL22  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
THIS IS AN SOS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


        100
 INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS INTO REGISTERS
**FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

        2475
**SECOND INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 010 ###


        9900
**THIRD INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 020 ###


          16
**FOURTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 030 ###


2147483647
**FIFTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 040 ###


90      -2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 090 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 104 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY. ADDITIONAL CREDIT IS GIVEN BELOW.**
        $$$$ADDITIONAL CREDIT: TOTAL MACHINE GRADE WITH BONUS POINTS= 100 $$$$
```

```
NAME: BACONR23  JCBNAME: EXAMPL23  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
THIS IS AN SCS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


       100
 INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS INTO REGISTERS
**FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      2476
 INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X)
**SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      9900
**THIRD INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 010 ###


        16
**FOURTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 020 ###


2147483647
**FIFTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 030 ###

91
-2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 080 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 105 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED FEWER INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY. ADDITIONAL CREDIT IS GIVEN BELOW.**
        $$$$ADDITIONAL CREDIT: TOTAL MACHINE GRADE WITH BONUS POINTS= 090 $$$$
```

```
NAME: BACONR24  JCBNAME: EXAMPL24  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
THIS IS AN SOS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


        99
**FIRST INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 010 ###


       2475
**SECOND INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 020 ###


       9900
**THIRD INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 030 ###


         16
**FOURTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 040 ###


2147483647
**FIFTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 050 ###

92

-2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 100 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 109 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**
```

```
NAME: BACONR25  JOBNAME: EXAMPL25  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360     29 MAY, 1969
THIS IS AN SCS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


      100
 INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS INTO REGISTERS
**FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      2476
 INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X)
**SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      9901
 THE BEAUTY OF USING SHIFT INSTRUCTIONS INSTEAD OF MULTIPLICATION AND DIVISION BY 2**N IS THAT IT ONLY INVOLVESONE REGISTER.
**THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

         16
**FOURTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 010 ###


2147483647
**FIFTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 020 ###

93

-2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 070 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 106 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**
```

```
NAME: BACONR26  JOBNAME: EXAMPL26  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360     29 MAY, 1969
THIS IS AN SCS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


      100
 INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS INTO REGISTERS
**FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

     2476
 INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X)
**SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

     9901
 THE BEAUTY OF USING SHIFT INSTRUCTIONS INSTEAD OF MULTIPLICATION AND DIVISION BY 2**N IS THAT IT ONLY INVOLVESONE REGISTER.
**THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

       17
 IS THE INSTRUCTION M 16,LABEL VALID?
**FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

2147483647
**FIFTH INTERMEDIATE ANSWER CORRECT**
    ###ACCUMULATED INTERMEDIATE ANSWER CREDIT= 010 ###

94
-2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 060 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 107 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**
```

```
NAME: BACONR27  JOBNAME: EXAMPL27  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360     29 MAY, 1969
THIS IS AN SOS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


      100
 INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS INTO REGISTERS
**FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      2476
 INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X)
**SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      9901
 THE BEAUTY OF USING SHIFT INSTRUCTIONS INSTEAD OF MULTIPLICATION AND DIVISION BY 2**N IS THAT IT ONLY INVOLVESONE REGISTER.
**THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      17
 IS THE INSTRUCTION M 16,LABEL VALID?
**FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

2147483646
 WHY IS THE ABSOLUTE VALUE OF THIS ANSWER LESS THAN THE ABSOLUTE VALUE OF THE NEXT ANSWER?
**FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

-2147483648
**FINAL ANSWER CORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 050 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: 109 INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**
```

```
NAME: BACONR28  JOBNAME: EXAMPL2B  NAVAL POSTGRAD SCHOOL STUDENT OPERATING SYSTEM  SOS/360    29 MAY, 1969
THIS IS AN SCS GRADER RUN -- GOOD LUCK
**PROB. 1 INITIALIZATION BEGUN**
**INITIALIZATION SUCCESSFUL**


      100
 INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS INTO REGISTERS
**FIRST INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      2476
 INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X)
**SECOND INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      5901
 THE BEAUTY OF USING SHIFT INSTRUCTIONS INSTEAD OF MULTIPLICATION AND DIVISION BY 2**N IS THAT IT ONLY INVOLVES ONE REGISTER.
**THIRD INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

      17
 IS THE INSTRUCTION M 16,LABEL VALID?
**FOURTH INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

2147483646
 WHY IS THE ABSOLUTE VALUE OF THIS ANSWER LESS THAN THE ABSOLUTE VALUE OF THE NEXT ANSWER?
**FIFTH INTERMEDIATE ANSWER NOT CORRECT. DATA SUBSTITUTED. ROUTINE CONTINUED.**

96
-2147483647
**FINAL ANSWER INCORRECT**
        $$$$END OF GRADER RUN: TOTAL MACHINE GRADE= 050 $$$$
 INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OF A DECIMAL CONSTANT
**HALT INSTRUCTION: ICS INSTRUCTIONS EXECUTED**
**YOUR PROGRAM REQUIRED MORE INSTRUCTIONS THAN THE MEDIAN FOR PROJECTS IN THIS CATEGORY.**
```

APPENDIX E

Example of Instructor's Answer Control Section.

*** APPENDIX E: INSTRUCTOR'S ANSWER CONTROL SECTION ***

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    3          MACRO
                                    4 &NAME    ANS
                                    5          LCLA  &N
                                    6 .*
                                    7 .*************************************************************************
                                    8 .*************************************************************************
                                    9 .**      THIS MACRO WILL DEFINE THE SIX ANSWERS REQUIRED BY THE      **
                                   10 .**      GRADER.  THE OPERANDS FOR 'ANS' ARE: FIRST ANSWER, SECOND   **
                                   11 .**      ANSWER,....FIFTH ANSWER,FINAL ANSWER.  THE ANSWERS ARE      **
                                   12 .**      SEPARATED BY COMMAS AND MUST BE DEFINED IN THE ABOVE        **
                                   13 .**      ORDER.  THE FOLLOWING IS AN EXAMPLE:                        **
                                   14 .**      ANS   100,120,99,-450      DEFINES THE FIRST THREE          **
                                   15 .**      INTERMEDIATE ANSWERS AND A FINAL ANSWER (-450).             **
                                   16 .**      INTERMEDIATE ANSWERS 4 AND 5 WILL BE DEFINED AS ZERO.       **
                                   17 .*************************************************************************
                                   18 .*************************************************************************
                                   19 .*
                                   20          AIF   (N'&SYSLIST EQ 0).ERROR
                                   21          AIF   (N'&SYSLIST GT 6).ERROR
                                   22          AIF   (N'&SYSLIST EQ 1).LAST
                                   23 .LOOP    ANOP
                                   24 &N       SETA  &N+1
                                   25 ANS&N    DC    F'&SYSLIST(&N)'
                                   26          AIF   (&N LT N'&SYSLIST-1).LOOP
                                   27          AIF   (&N EQ 5).LAST
                                   28 .LOOP1   ANOP
                                   29 &N       SETA  &N+1
                                   30 ANS&N    DC    F'0'
                                   31          AIF   (&N LT 5).LOOP1
                                   32 .LAST    ANOP
                                   33 &N       SETA  N'&SYSLIST
                                   34 ANSF     DC    F'&SYSLIST(&N)'
                                   35          AGO   .LAST1
                                   36 .ERROR   MNOTE *,' ***INCORRECT NUMBER OF ANSWERS SPECIFIED***'
                                   37 .LAST1   ANOP
                                   38          MEND
```

*** APPENDIX E: INSTRUCTOR'S ANSWER CONTROL SECTION ***

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                   41          MACRO
                                   42 &NAME    PROF
                                   43          LCLA  &N,&B1,&B2,&B3
                                   44 .*
                                   45 .*********************************************************************
                                   46 .*********************************************************************
                                   47 .**      THIS MACRO SELECTS THE GRADER OPTIONS FOR THE INSTRUCTOR    **
                                   48 .**      A BLANK OPERAND FIELD SPECIFIES THAT ALL GRADER OPTIONS     **
                                   49 .**      ARE TO BE USED.  'NOMSG' SPECIFIES THAT NO INSTRUCTOR      **
                                   50 .**      MESSAGES ARE TO BE SENT.  'NODATA' SPECIFIES THAT CORRECT   **
                                   51 .**      DATA WILL NOT BE SUBSTITUTED FOR INCORRECT INTER-           **
                                   52 .**      MEDIATE ANSWERS.  'NOBONUS' SPECIFIES THAT THE CHECK OF     **
                                   53 .**      INSTRUCTIONS EXECUTED IS NOT TO BE MADE AND BONUS POINTS    **
                                   54 .**      ARE NOT ASSIGNED.  THE FOLLOWING IS AN EXAMPLE:             **
                                   55 .**      PROF  NOMSG,NOBONUS          NO MESSAGES ARE SENT,          **
                                   56 .**      NO BONUS IS GIVEN, AND CORRECT DATA IS SUBSTITUTED.         **
                                   57 .**      NOTE:  OPERANDS MAY BE PLACED IN ANY ORDER.                 **
                                   58 .*********************************************************************
                                   59 .*********************************************************************
                                   60 .*
                                   61          AIF   (N'&SYSLIST GT 3).ERROR1
                                   62 &B1      SETA  1
                                   63 &B2      SETA  1
                                   64 &B3      SETA  1
                                   65          AIF   (N'&SYSLIST EQ 0).HERE
                                   66 .LOOP    ANOP
                                   67 &N       SETA  &N+1
                                   68          AIF   ('&SYSLIST(&N)' EQ 'NOMSG').SET1
                                   69          AIF   ('&SYSLIST(&N)' EQ 'NODATA').SET2
                                   70          AIF   ('&SYSLIST(&N)' EQ 'NOBONUS').SET3
                                   71          AIF   (&N LE 2).LOOP
                                   72          AGO   .HERE
                                   73 .SET1    ANOP
                                   74 &B1      SETA  0
                                   75          AGO   .LOOP
                                   76 .SET2    ANOP
                                   77 &B2      SETA  0
                                   78          AGO   .LOOP
                                   79 .SET3    ANOP
                                   80 &B3      SETA  0
                                   81          AGO   .LOOP
                                   82 .ERROR1  MNOTE *,'ERROR, NO CONSTANT GENERATED'
                                   83          MEXIT
                                   84 .HERE    ANOP
                                   85 CONTROL  DC    X'FF'
                                   86          DC    B'1111&B1&B2&B3.1'
                                   87          MEND
```

*** APPENDIX E: INSTRUCTOR'S ANSWER CONTROL SECTION ***

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                   90          MACRO
                                   91 &GNAME   IMSG  &MSG
                                   92 .*
                                   93 .*************************************************************************
                                   94 .*************************************************************************
                                   95 .**       THIS MACRO WILL DEFINE MESSAGES TO THE STUDENT.  THEY       **
                                   96 .**       MUST BE PLACED IN THE FOLLOWING ORDER:  NAMES: IMESS1,      **
                                   97 .**       IMESS2,IMESS3,IMESS4,IMESS5,IMESSF.  THE OPERAND OF 'IMSG' **
                                   98 .**       IS A CHARACTER STRING ENCLOSED IN APOSTROPHES UP TO 132     **
                                   99 .**       CHARACTERS LONG.  THE FOLLOWING IS AN EXAMPLE:             **
                                  100 .**       IMESS1   IMSG   ' THIS IS INSTRUCTOR MESSAGE NO. 1 '       **
                                  101 .*************************************************************************
                                  102 .*************************************************************************
                                  103 .*
                                  104          ORG   *+2
                                  105 &GNAME.M DS    0CL132
                                  106          ORG   *-2
                                  107 &GNAME   DC    AL1(131)
                                  108          DC    XL1'02'
                                  109          DC    CL132&MSG
                                  110          ORG
                                  111          MEND
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000000                           114 GINITC   CSECT
                                 115 PROFS    ANS   99,2475,9900,16,2147483647,-2147483648
000000 00000063                  116+ANS1     DC    F'99'
000004 00009AB                   117+ANS2     DC    F'2475'
000008 000026AC                  118+ANS3     DC    F'9900'
00000C 00000010                  119+ANS4     DC    F'16'
000010 7FFFFFFF                  120+ANS5     DC    F'2147483647'
000014 80000000                  121+ANSF     DC    F'-2147483648'
000018 00000005                  122 NOINTANS DC    F'5'
00001C 00000069                  123 NOX      DC    F'105'
                                 124 PROFS    PROF
000020 FF                        125+CONTROL  DC    X'FF'
000021 FF                        126+         DC    B'11111111'
                                 127 IMESS1   IMSG  ' INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTANTS IX
                                                    NTO REGISTERS '
000024                           128+         ORG   *+2
000024                           129+IMESS1M  DS    0CL132
000022                           130+         ORG   *-2
000022 83                        131+IMESS1   DC    AL1(131)
000023 02                        132+         DC    XL1'02'
                                 133+         DC    CL132' INSTRUCTOR MESSAGE NO. 1: USE SXAI TO LOAD CONSTAX
000024 4CC9D5E2E3D9E4C3                +              NTS INTO REGISTERS '
0000A8                           134+         ORG
                                 135 IMESS2   IMSG  ' INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X) '
0000AA                           136+         ORG   *+2
0000AA                           137+IMESS2M  DS    0CL132
0000A8                           138+         ORG   *-2
0000A8 83                        139+IMESS2   DC    AL1(131)
0000A9 02                        140+         DC    XL1'02'
0000AA 40C9D5E2E3D9E4C3          141+         DC    CL132' INSTRUCTOR MESSAGE NO. 2: BCT R,LOC(X) '
00012E                           142+         ORG
                                 143 IMESS3   IMSG  ' THE BEAUTY OF USING SHIFT INSTRUCTIONS INSTEAD OF MULTX
                                                    IPLICATION AND DIVISION BY 2**N IS THAT IT ONLY INVOLVESX
                                                     ONE REGISTER. '
101 000130                       144+         ORG   *+2
000130                           145+IMESS3M  DS    0CL132
00012E                           146+         ORG   *-2
00012E 83                        147+IMESS3   DC    AL1(131)
00012F 02                        148+         DC    XL1'02'
                                 149+         DC    CL132' THE BEAUTY OF USING SHIFT INSTRUCTIONS INSTEAD OFX
                                    +               MULTIPLICATION AND DIVISION BY 2**N IS THAT IT ONLY INVX
000130 40E3C8C540C2C5C1          150+               OLVES ONE REGISTER. '
000184                           151+         ORG
                                 152 IMESS4   IMSG  ' IS THE INSTRUCTION M 16,LABEL VALID? '
000186                           153+         ORG   *+2
000186                           154+IMESS4M  DS    0CL132
000184                           155+         ORG   *-2
000184 83                        156+IMESS4   DC    AL1(131)
000185 02                        157+         DC    XL1'02'
000186 40C9E240E3C8C540          158+         DC    CL132' IS THE INSTRUCTION M 16,LABEL VALID? '
00023A                           159+         ORG
                                 160 IMESS5   IMSG  ' WHY IS THE ABSOLUTE VALUE OF THIS ANSWER LESS THAN THEX
                                                     ABSOLUTE VALUE OF THE NEXT ANSWER? '
00023C                           161+         ORG   *+2
00023C                           162+IMESS5M  DS    0CL132
00023A                           163+         ORG   *-2
00023A 83                        164+IMESS5   DC    AL1(131)
00023B 02                        165+         DC    XL1'02'
                                 166+         DC    CL132' WHY IS THE ABSOLUTE VALUE OF THIS ANSWER LESS THAX
00023C 40E6C8E840C9E240             +              N THE ABSOLUTE VALUE OF THE NEXT ANSWER? '
```

*** APPENDIX E: INSTRUCTOR'S ANSWER CONTROL SECTION ***

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
00C2C0                             166+           ORG
                                   167 IMESSF     IMSG  ' INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALUE OFX
                                                        A DECIMAL CONSTANT '
00C2C2                             168+           ORG   *+2
00C2C2                             169+IMESSFM    DS    0CL132
00C2C0                             170+           ORG   *-2
00C2C0 83                          171+IMESSF     DC    AL1(131)
00C2C1 02                          172+           DC    XL1'02'
00C2C2 40C9D5E2E3D9E4C3            173+           DC    CL132' INSTRUCTOR FINAL MESSAGE: THIS IS THE MINIMUM VALX
00C346                                +                  UE OF A DECIMAL CONSTANT '
                                   174+           ORG
                                   175            END
```

BIBLIOGRAPHY

1. Hollingsworth, Jack, "Automatic Graders for Programming Classes," Communications of the ACM, v. 3, p. 528-529, October 1960.

2. Forsythe, George E., and Wirth, Nicklaus, "Automatic Grading Programs," Communications of the ACM, v. 8, p. 275-278, May 1965.

3. Forsythe, George E., Automatic Machine Grading Systems, Proceedings A. C. M. National Conference, 1964, p. N1.4-1, A.C.M., 1964.

4. Berry, R. E., "Grader Programs," The Computer Journal, v. 9, p. 252-256, 1966.

5. Naur, P., "Automatic Grading of Students ALGOL Programming," B.I.T., v. 4, p. 177, 1964.

6. Temperley, J. F., and Smith, Barry W., "A Grading Procedure for PL/1 Student Exercises," The Computer Journal, v. 10, p. 368-374, February 1968.

7. Hext, J. B., and Winings, J. W., "An Automatic Grading Scheme for Simple Programming Exercises," Communications of the ACM, v. 12, p. 272-275, May 1969.

8. Montalbano, Michael S., Expressing Program Logic, Data Processing Digest, v. 14, p. 1-16, September 1968.

9. Opler, Ascher, Is Assembly Language Programming Passe' ?, Data Processing Digest, v. 14, p. 1-15, October 1968.

10. Wile, David S., Munck, Robert G., and van Dam, Andries, The Brown University Student Operating System, Proceedings of 22nd National Conference, A.C.M., p. 427-439, Thompson Book Company, 1967.

11. Brown University Students in A. M. 101-102, SOS Brown University Student Operating System for IBM System/360 Model 50, Center for Computer and Information Sciences, Brown University, November, 1968.

INITIAL DISTRIBUTION LIST

		No Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chief of Naval Operations (OP-91) Department of the Navy Washington, D.C. 20350	1
4.	Prof. G. L. Barksdale, Jr. (thesis advisor) Department of Mathematics (Code 53Bv) Naval Postgraduate School Monterey, California 93940	1
5.	Prof. Edward A. Singer Department of Mathematics (Code 53Sf) Naval Postgraduate School Monterey, California 93940	1
6.	Prof. W. S. Brainerd Department of Mathematics (Code 53Bz) Naval Postgraduate School Monterey, California 93940	1
7.	Prof. D. G. Williams Computing Center (Code 0211) Naval Postgraduate School Monterey, California 93940	1
8.	Prof. Andries van Dam Brown University Providence, Rhode Island 02912	1
9.	Richard M. Kogut Computer Laboratory Brown University Providence, Rhode Island 02912	1
10.	LCDR Roger F. Bacon, USN (student) USS HALIBUT (SSN-587) FPO San Francisco 96601	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Naval Postgraduate School
Monterey, California

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
An Investigation of An Automatic Machine Grading System for Assembly Language Instruction

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*
Master's Thesis; (June 1969)

5. AUTHOR(S) *(First name, middle initial, last name)*
Roger Francis Bacon

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1969	105	11

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

Machine grading techniques are becoming of greater importance because of the increased number of students in programming classes. Characteristics and limitations of automatic machine grading systems are proposed. A grader for introductory assembly language programming courses was developed. The properties of this grader are discussed and an example of a graded program is given.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Grader Program Characteristics						
Automatic Machine Grading						
Assembly Language Grading						
Grader Programs						
Computer Grading Programs						

Thesis 109295
B115 Bacon
c.1 An investigation of an automatic machine grading system for assembly language instruction.